60000287

CONTROL DATA

CYBER 96X MSL MAINTENANCE SOFTWARE REFERENCE MANUAL

MCT3/P TEST DESCRIPTION

CDC® COMPUTER SYSTEMS
CYBER 960, 962

PREFACE

This manual describes the CDC CYBER 960 and 962 Computer Systems MCT3/P test. Procedures for using this test are provided in the CYBER 96X Test Procedures Reference manual.

NOTE

It is assumed and planned that the user of program descriptions will have an in-depth understanding of the processor logic; including familiarity with all levels of the Multilevel Block Diagrams.

DISCLAIMER

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undefined parameters.

REVISION RECORD

REVISION	DESCRIPTION
A (07-11-88)	Manual released.

Publication No.
60000287

REVISION LETTERS I, O, Q, S, X, AND Z ARE NOT USED

1988

by Control Data Corporation
Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Technology and Graphics Division
4201 Lexington Avenue North
St. Paul, Minnesota 55126-6198

or use Comment Sheet in the back of this manual.

CONTENTS

MCT3/P MAINTENANCE ACCESS CONTROL (MAC) TEST 1
General 1
Section Descriptions 5
Section 0 - Initial MCH/MAC Communication Check 5
Section 0, Subsection 0 7
Section 0, Subsection 1 8
Section 0, Subsection 2 10
Section 0, Subsection 3 11
Section 0, Subsection 4 11
Section 0, Subsection 5 12
Section 1 - Initial Check of MAC Writing and Reading Capability 14
Section 1, Subsection 0 15
Section 1, Subsection 1 18
Section 1, Subsection 2 20
Section 1, Subsection 3 23
Section 1, Subsection 4 25
Section 1, Subsection 5 27
Section 1, Subsection 6 29
Section 1, Subsection 7 31
Section 1, Subsection 8 33
Section 1, Subsection 9 34
Section 1, Subsection 10 35
Section 1, Subsection 11 36
Section 1, Subsection 12 36
Section 1, Subsection 13 37
Section 1, Subsection 14 38
Section 1, Subsection 15 39
Section 2 - Test MAC Data Bus Paths 40
Section 2, Subsection 0 40
Section 2, Subsection 1 43
Section 2, Subsection 2 45
Section 2, Subsection 3 47
Section 3 - Test PFS Register Boards 50
Section 3, Subsection 0 51
Section 3, Subsection 1 52
Section 3, Subsection 2 54
Section 3, Subsection 3 56
Section 3, Subsection 4 59
Section 4 - Test MAC Hardware Error Reporting 61
Section 4, Subsection 0 62
Section 4, Subsection 1 62
Section 4, Subsection 2 64
Section 4, Subsection 3 65
Section 4, Subsection 4 66
Section 4, Subsection 5 67
Section 4, Subsection 6 68
Section 4, Subsection 7 69
Section 4, Subsection 8 70
Section 4, Subsection 9 71

Section 5 - Test MAC Hardware Parity Checkers 73
Section 5, Subsection 0 73
Section 5, Subsection 1 74
Section 6 - Test MAC Reference ROM Memory 77
Section 6, Subsection 0 77
Section 6, Subsection 1 78
Section 6, Subsection 2 79
Section 7 - Test MAC Address Bus Path 82
Section 7, Subsection 0 82
Section 8 - Test AC/ALN SCM PAKS Error Reporting and Byte Selects 85
Section 8, Subsection 0 86
Section 8, Subsection 1 87
Section 8, Subsection 2 88
Section 9 - Test BDP RAM PAKS Error Reporting and Byte Selects 90
Section 9, Subsection 0 91
Section 9, Subsection 1 92
Section 9, Subsection 2 93
Section 9, Subsection 3 94
Section 9, Subsection 4 95
Section 9, Subsection 5 97
Section 10 - Test IF Decode RAMS Error Reporting and Byte Selects 98
Section 10, Subsection 0 99
Section 10, Subsection 1 100
Section 11 - Test MAC Address Bus Error Reporting 102
Section 11, Subsection 0 103
Section 11, Subsection 1 104
Section 11, Subsection 2 106
Section 11, Subsection 3 107
Section 12 - Test IF C180 and C170 Decode RAM Memories 109
Section 12, Subsection 0 110
Section 12, Subsection 1 111
Section 12, Subsection 2 112
Section 12, Subsection 3 114
Section 12, Subsection 4 115
Section 12, Subsection 5 116
Section 12, Subsection 6 118
Section 12, Subsection 7 121
Section 12, Subsection 8 122
Section 13 - Test AC Soft Control 1 RAM Memories 125
Section 13, Subsection 0 125
Section 13, Subsection 1 127
Section 13, Subsection 2 129
Section 13, Subsection 3 132
Section 14 - Test AC Soft Control 2 RAM Memories 135
Section 14, Subsection 0 135
Section 14, Subsection 1 137
Section 14, Subsection 2 139
Section 14, Subsection 3 141
Section 15 - Test ALN SC RAM Memories 145
Section 15, Subsection 0 145
Section 15, Subsection 1 147
Section 15, Subsection 2 148
Section 15, Subsection 3 152

Section 16 - Test BDP Port A Memory Board 155
Section 16, Subsection 0 156
Section 16, Subsection 1 157
Section 16, Subsection 2 158
Section 16, Subsection 3 160
Section 17 - Test BDP Port B Memory Board 163
Section 17, Subsection 0 164
Section 17, Subsection 1 165
Section 17, Subsection 2 166
Section 17, Subsection 3 168
Section 18 - Test BDP RAM PAK 0 EBCDIC Memory 171
Section 18, Subsection 0 173
Section 18, Subsection 1 173
Section 18, Subsection 2 174
Section 18, Subsection 3 176
Section 19 - Test BDP RAM PAK 0 CONVERT Memory 179
Section 19, Subsection 0 180
Section 19, Subsection 1 181
Section 19, Subsection 2 182
Section 19, Subsection 3 184
Section 20 - Test BDP RAM PAK 1 SPEC ERROR Memory 187
Section 20, Subsection 0 188
Section 20, Subsection 1 189
Section 20, Subsection 2 190
Section 20, Subsection 3 192
Section 21 - Test BDP RAM PAK 1 X256 Memory 195
Section 21, Subsection 0 196
Section 21, Subsection 1 197
Section 21, Subsection 2 198
Section 21, Subsection 3 200
Section 22 - Test Register File 203
Section 22, Subsection 0 204
Section 22, Subsection 1 205
Section 22, Subsection 2 207
Section 22, Subsection 3 209
Section 23 - Test Control Store Breakpoint and Microand Address Registers 213
Section 23, Subsection 0 214
Section 23, Subsection 1 215
Section 23, Subsection 2 216
Section 23, Subsection 3 218
Section 23, Subsection 4 218
Section 24 - Test Control Store Memory 221
Section 24, Subsection 0 223
Section 24, Subsection 1 225
Section 24, Subsection 2 227
Section 24, Subsection 3 230
Section 25 - Complete Test of MAC Byte Select Logic 233
Section 25, Subsection 0 233
Section 25, Subsection 1 235
Section 25, Subsection 2 236
Section 26 - Complete Test of MAC Type Decode Logic 238
Section 26, Subsection 0 238

Section 27 - Complete Test of MAC Go Decoder 242
Section 27, Subsection 0 243
Section 27, Subsection 1 245
Section 28 - Miscellaneous MAC Logic Testing 248
Section 28, Subsection 0 248
Section 28, Subsection 1 250
Section 28, Subsection 2 251
Section 29 - test FFC logic 253
Section 29, Subsection 0 253
Section 29, Subsection 1 254

MAINTENANCE ACCESS CONTROL (MAC) TEST, MCT3/P

==========

GENERAL

MCT3/P tests the processor MAC logic and provides information that can be used to aid in the isolation of faulty MAC hardware. In addition, MCT3/P tests the individual internal memories and provides information that can be used to aid in the isolation of faulty processor memory hardware.

MCT3/P consists of the following:

- Section 00 tests the IOU MCH to processor MAC communication paths, including related control logic.
- Sections 01 through 07 test the MAC logic, including the MAC Bus (Data and Address), the MAC Reference ROM memory RAMs, MAC error reporting, the clock delay system and the PFS Register boards.
- Sections 08 through 11 test the error reporting logic on the individual processor internal memory boards (AC and ALN SCM PAKS, BDP A and B Port memories, BDP RAM PAKS, IF Decode RAMs).
- Section 12 through 21 test the individual processor internal memory RAMS and their related addressing and control logic (AC and ALN SCM, BDP A and B Ports, BDP RAMs, IF Decode RAMs).
- Section 22 tests the processor Register File logic, including error reporting, data holding, and addressing logic.
- Sections 23 and 24 test the processor Control Store Memory, including the Control Store Microrand Address Register, Breakpoint Register, RAM memory chips, and addressing logic.
- Sections 25 through 29 complete the testing of processor MAC logic. That MAC logic that has not been completely tested by previous subsections.

Sections and subsections must be executed sequentially since valid isolation information depends on the successful completion of previous sections and subsections.

All MAC activity is controlled from the IOU PP. The PP program (MCT3/P) sends commands and data to MAC over the MCH and reads the contents of Maintenance Registers, Register File, and Control Store back via the MCH. The PP program makes all determinations as to whether the processor MAC logic is operating correctly and reports them to the CE via the IOU display console.

Channel Error Status Reporting:

Throughout the first few sections of MCT3/P, checks are performed to verify proper MCH operation and MAC/MCH operation and data transfers. Such operation consists of Active, Inactive, Full, and Empty states of the MCH and channel. The following format will be used to report one or more state errors and parity errors within one 64-bit Condition Result word:

byte	0	1	2	3	4	5	6	7
	Act.	Inact.	Full	Empty	Cnd No	FS 2/5	FS 2/6	FS 2/7

Act. = This byte position will be set to AA (hex) whenever an inactive MCH condition is expected but does not exist (MCH is active and should be inactive).

Inact. = This byte position will be set to 11 (hex) whenever an active MCH condition is expected but does not exist (MCH is inactive and should be active).

Full = This byte position will be set to FF (hex) whenever an empty MCH condition is expected but does not exist (MCH is full and should be empty).

Empty = This byte position will be set to EE (hex) whenever a full MCH condition is expected but does not exist (MCH is empty and should be full).

Cnd No = May contain the number of the first condition that sensed a chan error when chan error status is reported at the end of a subsection.

FS 2/5, FS 2/6 and FS 2/7 = Whenever a MCH channel error is detected, bytes 5, 6, and 7 of the IOU Fault Status Register 2 will be copied to bytes 5, 6, and 7 of the condition result. Bit 40 means channel 17 error (MCH), bit 57 means MCH Radial Interface 5/6 error, bit 58 means MCH Radial Interface 3/4 error, bit 59 means MCH Radial Interface 1/2 error. The other bits of PFS 2 bytes 5, 6, and 7 should have no effect on the MCH channel error, whether set or not.

STATUS report formats.
The following status formats are used in Section 1, subsections 10-15.

Maintenance channel status (report format 1)

2**11 - Chan ACTIVE prior to function
2**10 - CH ERROR during the transfer
2**09 - CH transfer terminated early
2**08 - CH did not go FULL
2**07 - CH ACTIVE dropped during transfer
2**06 - CH ERROR during address output
2**05 - CH remained FULL after output
2**04 - CH dropped ACTIVE during routine use
2**03 - not used.
2**02 - Channel FULL (No response to FCN)
2**01 - Channel ERROR (FCN PE)
2**00 - Channel not INACTIVE after function

CLOCK STATUS (Status report format 2)
The format of the status word is as follows -
(The bit number is the power of 2.)

Bit 16 PFS 81(16) - AC clock array parity error
Bit 15 PFS 81(17) - CMCB clock array parity error
Bit 14 PFS 81(18) - CSU clock array parity error 1
Bit 13 PFS 81(19) - CSU clock array parity error 2
Bit 12 PFS 83(16) - LMA clock array parity error
Bit 11 PFS 83(17) - LMB clock array parity error
Bit 10 PFS 83(18) - CMCA clock array parity error
Bit 09 PFS 85(16) - OPSM clock array parity error
Bit 08 PFS 85(17) - OPIB clock array parity error
Bit 07 PFS 85(18) - OPIA clock array parity error
Bit 06 PFS 85(19) - CSMC clock array parity error
Bit 05 PFS 87(16) - IFIC clock array parity error
Bit 04 PFS 87(17) - CSMC clock array parity error
Bit 03 PFS 87(18) - BPD clock array parity error
Bit 02 PFS 89(16) - ALN clock array parity error
Bit 01 PFS 89(17) - MDA clock array parity error
Bit 00 PFS 89(18) - MDB clock array parity error

Repeat Iteration:

Except where otherwise noted, setting Repeat Iteration (bit 2**13 in PARAM1) will cause the repetition of all conditions in a subsection.

Parity Error Status Reporting:

To provide isolation information when data errors are detected, the parity error bits are read from the PFS Registers and reported in one of the Subsection Condition Results. Since only a few of the over 400 parity error bits are of interest and valid at any one point in the test, the parity bits of interest are read from the different PFS Registers and formatted into the 64-bit Condition Result word. The formatting used varies from section to section, and even from subsection to subsection in some cases. The section/subsection documentation describes the particular format used in that section/subsection.

In all format definitions, undefined bit positions will always be zero.

Test Initialization:

- Clear Maintenance Channel error flag, if set.
- Clear IOU Fault Status Mask Register bits 40 and 59.
- Read the P3 and M3 DEC Registers and save the margin contents for later use. Also save the state of the First Failure Capture (FFC) enable bit.

SECTION DESCRIPTIONS

SECTION 0 - INITIAL MCH/MAC COMMUNICATION CHECK

This section verifies that the basic MAC control logic and MCH data paths operate correctly.

Subsection 0 - Tests that a Master clear function, Activate Channel and Deactivate Channel receive proper responses from the MAC logic.

Subsection 1 - Tests that an Echo sequence operates correctly. Only one data pattern used (FF (hex).

Subsection 2 - Tests the basic MCH data path from the IOU to MAC and from MAC back to the IOU using the Echo function with various data patterns.

Subsection 3 - Tests the MAC control logic specifically related to clearing the Control Word 1 FF and Control Word 2 FF.

Subsection 4 - Tests the MAC control logic specifically related to clearing the Control Words Loaded FF.

Subsection 5 - Tests that a Clear Errors Function sent to MAC receives proper responses from MAC logic.

Hardware Tested:

- The MCH control signal and data paths from the IOU to MAC.
- The MCH control signal and data paths from the MAC to IOU (excluding the data parity bit).
- MAC logic consisting of the MCH Data Receiver/Register, the MAC Address Register for Echo path, the Data Mux 3 (Maint. Chan. Adrs.), MCH Output Register and MCH Transmitters on the BZ MAC ZIF board.
- MAC logic consisting of the Ready, Function, Activate and Disconnect signals MAC channel receivers; control signal resync/gates; control word 1 and 2 logic; control words loaded logic; channel timing delay chain; Ready In and Disconnect signals to IOU. Most of this MAC logic is contained on the BZ MAC ZIF board, though a small portion of related logic is contained on the BK MAC ZIF board.

Method:

Basically, each subsection consists of a sequence of MCH operations such as Function, Deactivate, Activate, Output, and Input. Each operation is followed by a check(s) to ensure that the MAC logic circuitry responds with the proper signal(s) and data (where applicable).

Section 0, Subsection 0

This subsection tests that a Master Clear function, Activate Channel and Deactivate Channel receive proper responses from the MAC logic.

Method:

Using a Master Clear function (with type = 0 and type = A), this subsection verifies that the MAC will accept a function signal and return a disconnect in response to the function signal.

An Activate Channel instruction is then performed to check that the MAC Function Flags have cleared through the resync chain as they should. The MCH should stay active and empty approximately 100 milliseconds, at which time the MCH timeout logic should send back a disconnect which deactivates the MCH.

An Activate Channel instruction is then performed, followed immediately by a Deactivate instruction. The MCH should immediately go inactive and stay inactive and empty.

Conditions:

C.0 Check that the MCH is inactive and empty.
- a. Clear IOU Fault Status Registers and clear MCH channel error if set.
- b. Report error if MCH is not inactive and empty by setting Condition Result per Channel Error Status format; expected result equals zero.

C.1 Check MAC handling of Master Clear Function; Op Code equals six, Type Code equals zero.
- a. Send Master Clear Function; Op Code equals six, Type equals zero.
- b. Report error if MCH does not go inactive before timeout of 100 milliseconds by setting Condition Result to AA000--0 (hex); expected result equals zero.

C.2 Check that MCH is empty.
- a. Report error if MCH is full by setting Condition Result to 0000 FF00 0--0 (hex); expected result equals zero.

C.3 Check no channel errors on MCH.
- a. Report MCH channel error by storing IOU Fault Status Register 2 bytes 5, 6 and 7 in Condition Result; right justified; expected result equals zero.

C.4 Activate MCH and record time until MCH Timeout Disconnect, or 100.1 milliseconds.
- a. Report time until disconnect in microseconds; error if time is less than 90 or equals 200.

Conditions:

C.5 Activate MCH; wait 10 microseconds; Deactivate MCH and record time until MCH goes inactive, or 100.1 milliseconds.
a. Report time in microseconds until MCH goes inactive; error if time is greater then 10 (bytes 2, 3, and 4).

C.6 Send Master Clear Function; Op Code equals six, Type = A.
a. Check and report channel error status if MCH does not go inactive before timeout of 100 milliseconds by setting Condition Result to AA00--00 (hex); expected result equals zero.

C.7 Check that MCH is empty.
a. Report error if MCH is full by setting Condition Result to 0000 FF000--0 (hex); expected result equals zero.

C.8 Check no channel errors on MCH.
a. Report MCH channel error by storing IOU Fault Status Register 2 bytes 5, 6 and 7 in Condition Result; right justified; expected result equals zero.

Section 0, Subsection 1

This subsection tests that an Echo sequence operates correctly.

Method:

The subsection Master Clears the MAC, sends an Echo function, activates the MCH, outputs two Control Words (CW1 and CW2), deactivates the MCH, Activates the MCH, inputs first Echo byte and checks for correctness, inputs second Echo byte and checks for correctness, deactivates the MCH and exits subsection.

At each step in the above sequence of operations the MCH is checked for proper state (Active, Inactive, Empty and Full) as well as for any occurrences of channel parity errors.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
a. Report Channel Error Status; error if MCH is not inactive and empty after function sent; expected result equals zero.

C.1 Send Echo Function to MAC; Op Code equals eight, Type equals zero.
a. Report Channel Error Status; error if MCH is not inactive and empty after function sent; expected result equals zero.

Conditions:

C.2 Activate MCH.
a. Report Channel Error Status and error if MCH is not active and empty. Output first Control Word (CW1) equals FF (hex).
b. Report Channel Error Status (error if MCH is not active and empty; expected result equals zero).

C.3 Output second Control Word (CW2) equals FF (hex).
a. Report Channel Error Status; error if MCH is not active and empty; expected result equals zero.

C.4 Deactivate MCH.
a. Report Channel Error Status; error if MCH is not inactive and empty; expected result equals zero.

C.5 Activate MCH.
a. Report Channel Error Status; error if MCH does not go full and active; expected result equals zero.

C.6 Input one word from MCH.
a. Report Channel Error Status (error if MCH does not go full and stay active; expected result equals zero).

C.7 Report input word; error if not equal to FF (hex).

C.8 Input second word from MCH.
a. Report Channel Error Status (error if MCH does not go full and stay active; expected result equals zero).

C.9 Report input word; error if not equal to FF (hex).

C.10 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.11 If any MCH channel errors occurred, pick up IOU Fault Status Reg 2 bytes 5, 6 and 7 and number of first condition (byte 4) to sense a channel error and report Channel Error Status and error. (Expected condition result equals zero).

Section 0, Subsection 2

This subsection tests the basic MCH data path from the IOU to MAC and from MAC back to the IOU using the Echo function with various data patterns.

Method:

The subsection uses the echo function to send various data patterns to the MAC hardware. MAC should echo CW2 data back at which time the data is compared to what was sent. After each MCH operation (Function, Activate, Output, etc.), the MCH is also checked for proper state (Inactive, Empty, etc.). MCH channel parity errors are ignored.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

All patterns will be executed by looping through the following conditions.

Conditions:

C.0 Report number of pattern being used this pass in byte 1 of Condition Result, CW1 data in byte 5 and CW2 data (expected data) in byte 7 of Condition Result; informational.

C.1 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

C.2 Send Echo Function to MAC; Op Code equals eight, Type equals zero.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

C.3 Activate MCH.
a. Report Channel Error Status and error if MCH is not active and empty. Output first Control Word (CW1) equals CW1 Pattern.
b. Report Channel Error Status (error if MCH does not go empty and stay active or MCH channel error occurred; expected result equals zero).

C.4 Output second Control Word (CW2) equals CW2 Pattern.
a. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.5 Deactivate MCH.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

C.6 Activate MCH.
a. Report Channel Error Status (error if MCH does not go full and active; expected result equals zero).

C.7 Input one word from MCH.
a. Report Channel Error Status (error if MCH does not go full again and stay active; expected result equals zero).

C.8 Report input word; error if not equal to expected pattern.

C.9 Deactivate MCH.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

If Repeat Iteration set, go to C.1.

Exit subsection if end of operands.

Operand Patterns:

Hexadecimal data and pattern number.

Pattern No.	CW1	CW2 and Expected
0	00	FF
1	FF	00
2	00	AA
3	FF	AA
4	00	55
5	FF	55
6	00	CC
7	FF	CC
8	00	33
9	FF	33
A	00	10
B	FF	01
C	10	22
D	10	10

Section 0, Subsection 3

This subsection tests the MAC control logic specifically related to clearing the Control Word 1 FF and Control Word 2 FF.

Method:

The subsection initiates an Echo function, but only sends the first Control Word (CW1). The MCH is then deactivated by the test, then activated. At this point both Control Word 1 and Control Word 2 FFs should have been cleared (Control Word 1 FF Cleared by transmission of first Control Word, Control Word 2 FF cleared by the activate that followed the deactivate). Control Words Loaded FF should also be clear (never set).

The test then outputs one data word. The MAC hardware should be in such a state that MAC will not respond to this data word and the MCH should stay active and full (until MCH timeout). If either Control Word 1 or Control Word 2 FFs were not cleared, MAC will respond by making the MCH empty, which the test will report as an error.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
a. Report Channel Error Status if MCH is not inactive and empty after function sent; expected result equals zero.

C.1 Send Echo Function to MAC; Op Code equals eight, Type equals zero
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Activate MCH.
a. Report Channel Error Status and error if MCH is not active and empty. Output first Control Word (CW1) equals FF (hex).
b. Report Channel Error Status (error if MCH is not active and empty; expected result equals zero).

C.3 Deactivate MCH.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

C.4 Activate MCH
Output one data word of 0000 and record time until MCH goes empty, or 100.1 milliseconds.
a. Report time in microseconds until empty received.

Error if this time is less than 9999 or equals 100100 microseconds (bytes 1, 2, and 3).

C.5 Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero.

C.6 If MCH channel error occurred report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Section 0, Subsection 4

This subsection tests the MAC control logic specifically related to clearing the Control Words Loaded FF.

Method:

The subsection sends an Echo function, followed by both Control Words (CW1 and CW2), deactivates the MCH, and then sends a second Echo function followed by an Activate MCH. The second Echo function should have cleared the Control Words Loaded FF so that MAC hardware is waiting for two more control words. If MAC hardware has not cleared out the Control Words Loaded FF, MAC will output data when the MCH is activated the last time. This condition will be detected and reported as an error.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.1 Send Echo Function to MAC; Op Code equals eight, Type equals zero.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Activate MCH.
a. Report Channel Error Status and error if MCH is not active and empty. Output first Control Word (CW1) equals FF (hex).
b. Report Channel Error Status (error if MCH is not active and empty or MCH channel error occurred; expected result equals zero).

C.3 Output second Control Word (CW2) equals FF (hex).
a. Report Channel Error Status (error if MCH is not active and empty; expected result equals zero).

C.4 Deactivate MCH.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

C.5 Send Echo Function to MAC; Op Code equals eight, Type equals zero.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.6 Activate MCH.
a. Report Channel Error Status (error if MCH does not stay empty and active for at least 20 microseconds; expected result equals zero).

C.7 Deactivate MCH.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

C.8 If MCH channel error occurred, report error and contents of IOU Fault Status Register 2 bytes 5, 6, and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Section 0, Subsection 5

This subsection tests that a Clear Errors Function sent to MAC receives proper responses from MAC logic.

Method:

A single sequence of Master Clearing the MAC followed by sending a Clear Errors Function to MAC with Type equal to zero followed by sending a Clear Errors Function to MAC with Type equal to A is executed. A check is made after each function to ensure that MAC has responded correctly; inactive and empty.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
  a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.1 Check no channel errors on MCH.
  a. If channel error, report error and IOU Fault Status Register 2 bytes 5, 6, and 7 in Condition Result using Channel Error Status Format; expected result equals zero.

C.2 Send Clear Error Function to MAC; Op Code equals 7, Type equals zero.
  a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.3 Check no channel errors on MCH.
  a. If channel error, report error and IOU Fault Status Register 2 bytes 5, 6, and 7 using Channel Error Status Format; expected result equals zero.

C.4 Send Clear Error Function to MAC; Op Code equals 7, Type equals A.
  a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.5 Check no channel errors on MCH.
  a. If channel error, report error and IOU Fault Status Register 2 bytes 5, 6, and 7 using Channel Error Status Format; expected result equals zero.

SECTION 1 - INITIAL CHECK OF MAC WRITING AND READING CAPABILITY

This section will verify that MAC control logic related to writing and reading maintenance registers, Soft Control memories, Register File, and Control Store memory operates without hanging up. It will also ensure that the registers and memories have been written with good parity data.

Subsection 0 - This subsection tests that MAC will write all directly accessible Process Registers (Type 0) without hanging up.

Subsection 1 - This subsection tests that MAC will read all directly accessible Process Registers (Type 0) without hanging up.

Subsection 2 - This subsection tests that MAC will write all Soft Control Memories (Types 3, 4, 5, and 6) without hanging up.

Subsection 3 - This subsection tests that MAC will read (Sweep) all Soft Control Memories (Types 3, 4, 5, and 6) without hanging up.

Subsection 4 - This subsection tests that MAC will write all locations in the Register File (Type 7) without hanging up.

Subsection 5 - This subsection tests that MAC will read all locations in the Register File (Type 7) without hanging up.

Subsection 6 - This subsection tests that MAC will write all of Control Store Memory (Type 1) without hanging up.

Subsection 7 - This subsection tests that MAC will read all of Control Store Memory (Type 1) without hanging up.

Subsection 8 - This subsection tests the clearing of the Write FF by the function signal.

Subsection 9 - This subsection tests the clearing of the Read FF by the function signal.

Subsection 10 - This subsection tests that the clock PROM data can be read without error.

Subsection 11 - This subsection tests the MAC address bus and parity checkers applicable to the clock delay register system.

Subsection 12 - This subsection tests that the clock delays can be written and read without error. It also verifies that the registers can hold data.

Subsection 13 - This subsection tests that all clock delays can be read before any are read. Also verifies that reading of one delay does not disturb the data in another.

Subsection 14 - This subsection tests the parity detection and reporting of the clock fanout arrays.

Subsection 15 - This subsection tests the detection, by the clock fanout arrays, of addressing parity errors.

Hardware Tested:

- Control circuitry internal to the two MAC ZIF boards (BZ and BK) including type decode circuits, Write FFs, Read FFs, MAC Timing Chain, Channel Timing Chains.
- No hardware peculiar to the maintenance registers, Soft Control memories, Register File, nor Control Store memory is tested by this section.

Method:

Each subsection consists of a sequence of MCH operations required to write or read the maintenance registers, Soft Control Memory, Clock delay registers or Control Store Memory. After each MCH operation a check is made to verify that MAC has responded correctly to the particular operation. If MAC has not responded correctly an error is reported.

This section is also used to assure that all Maintenance Registers, Soft Control Memories, Control Store Memory and Register Files have been written with data with good parity. This assurance is required by later test sections.

Section 1, Subsection 0

This subsection tests that MAC will write all directly accessible Process Registers (Type 0) without hanging up.

Method:

The subsection tests MAC control hardware to ensure that no write Process Register operation hangs up because of bad circuitry. The function code and Control Word data is parity checked at the MAC channel receivers and any parity error will be reported as a MCH channel error. Data written is not checked.

The sequence of this subsection is as follows:

- Master Clear MAC to ensure ready state.
- Send Write Function.
- Activate MCH.
- Send Control Word 1 equal to zero and Control Word 2 equal to desired direct access Process Register Number.
- Deactivate MCH.
- Activate MCH.

- Send 16 bytes of zero data, checking for MCH active and empty after each byte.
- Report number of bytes sent. Should be 16 unless MCH did not go empty during output, then report would be number of bytes sent before MCH hangs up.
- Deactivate MCH.

After each operation, the state of the MCH is checked to ensure that MAC has responded properly (not hung up).

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

The following conditions are repeated for each directly accessible Processor State Register.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.

a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.1 Send Write Function to MAC; Op Code equals five, Type equals zero.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Activate MCH.
a. Report Channel Error Status and error if MCH is not active and empty. Output first Control Word (CW1) equals zero (0).
b. Report Channel Error Status (error if MCH is not active and empty; expected result equals zero).

C.3 Output second Control Word (CW2) equals Process Register Number being tested.
a. Report Channel Error Status (error if MCH is not active and empty; expected result equals zero).

C.4 Report Process Register Number being tested; informational.

C.5 Deactivate MCH.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

Conditions:

C.6 Activate MCH.
a. Report Channel Error Status and error if MCH is not active and empty, and skip steps b through e.
b. Initialize output byte count to 0.
c. Output data byte of 00 on MCH.
d. Report Channel Error Status (error if MCH is not active and empty; expected result equals zero).
e. Increment byte count (byte count plus one).

Repeat steps c through e if byte count not equal to 16 (10 (hex)) and no channel error occurred in step d.

C.7 Report number of bytes sent in C.6; error if not equal to 16 (10 (hex)).

C.8 Deactivate MCH.
a. Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

C.9 If MCH channel error occurred in C.0 through C.5; report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Clear IOU Fault Status Registers prior to repeating conditions.

Exit subsection if end of operands.

Operand Patterns:

Process Register Numbers (Hexadecimal) to be tested (used in CW2):

Pattern No.	Process Register No.
0	30
1	31
2	32
3	80
4	81
5	82
6	83
7	84
8	85
9	86
A	87
B	88
C	89
D	A0

Section 1, Subsection 1

This subsection tests that MAC will read all directly accessible Process Registers (Type 0) without hanging up.

Method:

The subsection tests MAC control hardware to ensure that no read Process Register operation hangs up because of bad circuitry. The function code and Control Word data is parity checked at the MAC channel receivers and any parity error will be reported as a MCH channel error. Data read is not checked.

The sequence of this subsection is as follows:

- Master Clear MAC to ensure ready state.
- Send Read Function.
- Activate MCH.
- Send Control Word 1 equal to zero, and Control Word 2 equal to desired direct access Process Register Number.
- Deactivate MCH.
- Activate MCH.
- Input 16 bytes of data, checking for MCH active and full after each byte. No data checking.
- Report number of bytes input. Should be 16 unless MCH did not go full during input, then report would be number of bytes sent by MAC before MCH hang up.
- Deactivate MCH.

After each operation, the state of the MCH is checked to ensure that MAC has responded properly (not hung up).

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

The following conditions are repeated for each directly accessible Processor State Register.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
- a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.1 Send Read Function to MAC; Op Code equals four, Type equals zero.
- a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Activate MCH.
- a. Report Channel Error Status and error if MCH does not go active and empty. Output first Control Word (CW1) equals zero.
- b. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.3 Output second Control Word (CW2) equal to Process Register Number being tested.
- a. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.4 Report Process Register Number being tested; informational.

C.5 Deactivate MCH.
- a. Check and report channel error status if MCH does not go inactive and empty; expected result equals zero).

C.6 Activate MCH.
- a. Report Channel Error Status and error if MCH does not go active and full, and skip steps b through e.
- b. Initialize input byte count to 0.
- c. Input one data byte (no data check).
- d. Report Channel Error Status (error if MCH does not go full and stay active; expected result equals zero).
- e. Increment byte count (byte count plus one).

Repeat steps c through e if byte count not equal to 16 (10 (hex) and no channel error occurred in step d.

C.7 Report number of bytes input in C.6. Error if not equal to 16 (10 (hex).

C.8 Deactivate MCH.
- a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.9 If MCH channel error occurred in C.0 through C.5; report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Clear IOU Fault Status Registers prior to repeating conditions.

Exit subsection if end of operands.

Operand Patterns:

Process Register Numbers (Hexadecimal) to be tested (used in CW2):

Pattern No.	Process Register No.
0	00
1	10
2	11
3	12
4	30
5	31
6	32
7	80
8	81
9	82
A	83
B	84
C	85
D	86
E	87
F	88
10	89
11	A0

Section 1, Subsection 2

This subsection tests that MAC will write all Soft Control Memories (types 3, 4, 5 and 6) without hanging up.

Method:

The subsection tests MAC control hardware to ensure that no write to a Soft Control Memory will hang up because of bad circuitry. The function code and Control Word data are parity checked at the MAC channel receivers and any parity error will be reported as a MCH channel error. Data written is not checked.

The sequence of this subsection is as follows:

- Master Clear MAC to ensure ready state.
- Send Write Function with type set to Soft Control Memory being written.
- Activate MCH.

Method:

- Send Control Word 1 equal to zero and Control Word 2 equal to zero. These values will start the memory write at location 0.
- Deactivate MCH.
- Activate MCH.
- Send a byte of zero data enough times to fill the particular memory (four bytes per location) checking for MCH active and empty after each byte.
- Report number of bytes sent. Should be four times the size of the memory unless MCH did not go empty during output; then report would be the number of bytes sent before MCH hung up.
- Deactivate MCH.

After each operation, the state of the MCH is checked to ensure that MAC has responded properly (not hung up).

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

The following conditions are repeated for each Soft Control Memory type.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
  a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.1 Send Write Function to MAC; Op code equals five, Type equals Soft Control Memory Type being written.
  a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Report Soft Control Memory Type being written in Condition Result byte 7 informational.

C.3 Activate MCH.
  a. Report Channel Error Status and error if MCH does not go active and empty. Output first Control Word (CW1) equals zero.
  b. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.4 Output second Control Word (CW2) equals zero.
  a. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

Conditions:

C.5 Deactivate MCH.
  a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.6 Activate MCH.
  a. Report Channel Error Status and error if MCH does not go active and stay empty, and skip steps b through e.
  b. Initialize output byte count to zero.
  c. Output data byte of zero on MCH.
  d. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).
  e. Increment byte count (byte count plus one).

  Repeat steps c through e if byte count does not equal four times the size of the type memory being written and no channel error occurred in step d.

C.7 Report the number of bytes sent in Condition 6. Error if byte count does not equal the following values for corresponding type: (Hexadecimal)

  Type 3 = 4 x 100 = 400
  Type 4 = 4 x 100 = 400
  Type 5 = 4 x 600 = 1800
  Type 6 = 4 x 200 = 800

C.8 Deactivate MCH.
  a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.9 If MCH channel error occurred in C.0 through C.5, report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Clear IOU Fault Status Registers prior to repeating conditions.

Exit subsection if end of memory types to write.

Memory Types to Write: (Hexadecimal)

Type Number	Address Range	Size
3	0 -- FF	100
4	0 -- FF	100
5	0 -- 5FF	600
6	0 -- 1FF	200

Section 1, Subsection 3

This subsection tests that MAC will read (sweep) all Soft Control Memories (types 3, 4, 5 and 6) without hanging up.

Method:

The subsection tests MAC control hardware to ensure that no read from a Soft Control Memory will hang up because of bad circuitry. The function code and Control Word data is parity checked at the MAC channel receivers and any parity error will be reported as a MCH channel error. Data read is not checked.

The sequence of this subsection is as follows:

- Master Clear MAC to ensure ready state.
- Send Read Function with type set to Soft Control Memory being written.
- Activate MCH.
- Send Control Word 1 equal to zero and Control Word 2 equal to zero. These values will start the memory read at location 0.
- Deactivate MCH.
- Activate MCH.
- Input a byte of data enough times to sweep the entire memory (one byte per location) checking for MCH active and full after each byte.
- Report number of bytes received. Should be equal to the size of the memory unless MCH did not go full during input; then report would be the number of bytes input before MCH hang up.
- Deactivate MCH.

After each operation, the state of the MCH is checked to ensure that MAC has responded properly (not hung up).

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

The following conditions are repeated for each Soft Control Memory type.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.1 Send Read Function to MAC; Op code equals four, Type equals Soft Control Memory Type being written.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Report soft Control Memory Type being read in Condition Result byte 7; informational.

C.3 Activate MCH.
a. Report Channel Error Status and error if MCH does not go active and empty. Output first Control Word (CW1) equals zero (0).
b. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.4 Output second Control Word (CW2) equals zero (0).
a. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.5 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.6 Activate MCH.
a. Report Channel Error Status and error if MCH does not go active and full, and skip steps b through e.
b. Initialize input byte count to 0.
c. Input data byte from MCH.
d. Report Channel Error Status (error if MCH does not go full and stay active; expected result equals zero).
e. Increment byte count (byte count plus one).

Repeat steps c through e if byte count not equal the size of the type memory being read and no channel error in step d.

C.7 Report the number of bytes input in Condition 6. Error if byte count is not equal the following values for corresponding type: (Hexadecimal)

Type 3 = 100
Type 4 = 100
Type 5 = 600
Type 6 = 200

C.8 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

Conditions:

C.9 If MCH channel error occurred in C.0 through C.5; report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Clear IOU Fault Status Registers prior to repeating conditions.

Exit subsection if end of memory types to read.

Memory Types to Read: (Hexadecimal)

Type No.	Address Range	Size
3	0 -- FF	100
4	0 -- FF	100
5	0 -- 5FF	600
6	0 -- 1FF	200

Section 1, Subsection 4

This subsection tests that MAC will write all locations in the Register File (Type 7) without hanging up.

Method:

The subsection tests MAC control hardware to ensure that no write to the Register File hangs up because of bad circuitry. The function code and Control Word data is parity checked at the MAC channel receivers and any parity error will be reported as a MCH channel error. Data written is not checked.

The sequence of the subsection is as follows:

- Master Clear MAC to ensure ready state.
- Send Write Function with type equal to seven.
- Activate MCH.
- Send Control Word 1 equal to zero and Control Word 2 equal to zero. This provides starting address of 0 in the Register File.
- Deactivate MCH.
- Activate MCH.
- Send 200 (hex) bytes of FF (hex) data, checking for MCH active and empty after each byte. FF (hex) data used to sensitize CS Disassembly Mux for later test sections.

- Report number of bytes sent. Should be 200 (hex) unless MCH did not go empty during output, then report would be number of bytes sent before MCH hung up.
- Deactivate MCH.

After each operation, the state of the MCH is checked to ensure that MCH has responded properly (not hung up).

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
  a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.1 Send Write Function to MAC; Op Code equals five, Type equals 7.
  a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Activate MCH.
  a. Report Channel Error Status and error if MCH does not go active and empty. Output first Control Word (CW1) equals zero.
  b. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.3 Output second Control Word (CW2) equals zero.
  a. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.4 Deactivate MCH.
  a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.5 Activate MCH.
  a. Report Channel Error Status and error if MCH does not go active and stay empty, and skip steps b through e.
  b. Initialize output byte count to 0.
  c. Output data byte of FF (hex) on MCH.
  d. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).
  e. Increment byte count (byte count plus one).

  Repeat steps c through e if byte count not equal 200 (hex) and no channel error occurred in step d.

C.6 Report number of bytes sent in C.5. Error if not equal to 200 (hex).

Conditions:

C.7 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.8 If MCH channel error occurred in C.0 through C.4; report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Section 1, Subsection 5

This subsection tests that MAC will read all locations in the Register File (Type 7) without hanging up.

Method:

The subsection tests MAC control hardware to ensure that no read from the Register File hangs up because of bad circuitry. The function code and Control Word data is parity checked at the MAC channel receivers and any parity error will be reported as a MCH channel error. Data read is not checked.

The sequence of the subsection is as follows:

- Master Clear MAC to ensure ready state.
- Send Read Function with type equal to seven.
- Activate MCH.

Method:

- Send Control Word 1 equal to zero and Control Word 2 equal to zero. This provides starting address of 0 in the Register File.
- Deactivate MCH.
- Activate MCH.
- Input 200 (hex) bytes of data, checking for MCH active and full after each byte.
- Report number of bytes input. Should be 200 (hex) unless MCH did not go full during input, then report would be number of bytes received before MCH hang up.
- Deactivate MCH.

After each operation, the state of the MCH is checked to ensure that MCH has responded properly (not hung up).

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0 Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.1 Send Read Function to MAC; Op Code equals four, Type equals seven.

a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Activate MCH.
a. Report Channel Error Status and error if MCH does not go active and empty. Output first Control Word (CW1) equals zero.
b. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.3 Output second Control Word (CW2) equals zero.
a. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.4 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.5 Activate MCH.
a. Report Channel Error Status and error if MCH does not go active and full, and skip steps b through e.
b. Initialize input byte count to 0.
c. Input data byte from MCH.
d. Report Channel Error Status (error if MCH does not go full and stay active; expected result equals zero).
e. Increment byte count (byte count plus one).

Repeat steps c through e if byte count not equal 200 (hex) and no channel error in d.

C.6 Report number of bytes input in C.5. Error if not equal to 200 (hex).

C.7 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.8 If MCH channel error occurred in C.0 through C.4; report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Section 1, Subsection 6

This subsection tests that MAC will write all of Control Store Memory (Type 1) without hanging up.

Method:

The subsection tests MAC control hardware to ensure that no write to Control Store Memory will hang up because of bad circuitry. The function code and Control Word data is parity checked at the MAC channel receivers and any parity error will be reported as a MCH channel error. Data written is not checked.

The sequence of this subsection is as follows:

- Master Clear MAC to ensure ready state.
- Write Control Store Address Register (31 (hex) with starting address of 0--0 (proper Activates, Functions, CWs, and Deactivates are used).
- Send Write function with Type equal to one.
- Activate MCH.
- Send dummy control words (CW1 and CW2) both equal to zero.
- Deactivate MCH.
- Activate MCH.
- Send 8000 (hex) bytes of FF (hex) data, checking for MCH active and empty after each byte. FF (hex) data is used to sensitize CS Disassembly Mux for later test sections.
- Report number of bytes sent. Should be 8000 (hex) unless MCH did not go empty during output, then report would be number of bytes sent before MCH hung up.
- Deactivate MCH.

After each operation, the state of the MCH is checked to ensure that MAC has responded properly (not hung up).

Initialization

- Clear IOU Fault Status Register and clear MCH channel error if set.

Conditions:

C.0 a. Master Clear MAC, check that MCH goes inactive and empty after function.

b. Write Process Register 31 (hex) (Control Store Address Register) with zeros, check for errors after each operation in the write sequence.
c. If any errors occurred in steps a and b, report appropriate Channel Error Status and report error; expected result equals zero.

C.1 Send Write Function to MAC; Op code equals five, Type equals one.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Activate MCH.
a. Report Channel Error Status and error if MCH does not go active and empty. Output first Control Word (CW1) equal to zero.
b. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.3 Output second dummy Control Word (CW2) equal to zero.
a. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.4 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.5 Activate MCH.
a. Report Channel Error Status and error if MCH does not go active and empty, and skip steps b through e.
b. Initialize output byte count to zero.
c. Output data byte of FF (hex) to MCH.
d. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).
e. Increment byte count (byte count plus one).

Repeat steps c through e if byte count not equal 8000 (hex) and no channel error occurred in step d.

C.6 Report number of bytes sent in C.5. Error if not equal to 8000 (hex).

C.7 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.8 If MCH channel error occurred in C.0 through C.4, report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Section 1, Subsection 7

This subsection tests that MAC will read all of Control Store Memory (Type 1) without hanging up.

Method:

The subsection tests MAC control hardware to ensure that no read from Control Store Memory will hang up because of bad circuitry. The function code and Control Word data is parity checked at the MAC channel receivers and any parity error will be reported as a MCH channel error. Data read is not checked.

The sequence of this subsection is as follows:

- Master Clear MAC to ensure ready state.
- Write Control Store Address Register (31 (hex) with starting address of 0--0 (proper Activates, Functions, CWs, and Deactivates are used).
- Send Read function with Type equal to one.
- Activate MCH.
- Send dummy control words (CW1 and CW2) both equal to zero.
- Deactivate MCH.
- Activate MCH.
- Input 8000 (hex) bytes of data, checking for MCH active and full after each byte.
- Report number of bytes input. Should be 8000 (hex) unless MCH did not go full during input then report would be number of bytes input before MCH hung up.
- Deactivate MCH.

After each operation, the state of the MCH is checked to ensure that MAC has responded properly (not hung up).

Initialization:

- Clear IOU Fault Status Register and clear MCH channel error if set.

Conditions:

C.0
a. Master Clear MAC, check that MCH goes inactive and empty after function.
b. Write Process Register 31 (hex) (Control Store Address Register) with zeros, check for errors after each operation in the write sequence.
c. If any errors occurred in steps a and b, report appropriate Channel Error Status and report error; expected result equals zero.

C.1 Send Read Function to MAC; Op code equals four, Type equals one.
a. Report Channel Error Status (error if MCH is not inactive and empty after function sent; expected result equals zero).

C.2 Activate MCH.
a. Report Channel Error Status and error if MCH does not go active and empty. Output first Control Word (CW1) equal to zero.

b. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.3 Output second dummy Control Word (CW2) equal to zero.
a. Report Channel Error Status (error if MCH does not go empty and stay active; expected result equals zero).

C.4 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.5 Activate MCH.
a. Report Channel Error Status and error if MCH does not go active and full, and skip steps b through e.
b. Initialize input byte count to zero.
c. Input data byte from MCH.
d. Report Channel Error Status (error if MCH does not go full and stay active; expected result equals zero).
e. Increment byte count (byte count plus one).

Repeat steps c through e if byte count not equal 8000 (hex) and no channel error in d.

C.6 Report number of bytes input in C.5. Error if not equal 8000 (hex).

C.7 Deactivate MCH.
a. Report Channel Error Status (error if MCH does not go inactive and empty; expected result equals zero).

C.8 If MCH channel error occurred in C.0 through C.4, report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition (byte 4) that sensed a channel error; expected result equals zero.

Section 1, Subsection 8

This subsection tests the clearing of the Write FF by the function signal.

Method:

The basic method used to verify that the Write FF clears when MAC functions are set is as follows:

- Perform a write function to the MAC Reference ROM Memory. This operation sets the Write FF (FF is on the BZ ZIF board at C1E).
- Send four bytes and then deactivate the MCH.
- Send a Master Clear function to MAC. The function signal should clear the Write FF.
- Activate the MCH, send two dummy Control Words to MAC, and then deactivate MCH.
- Activate the MCH and send one byte of data to MAC. A properly operating MAC should not respond to this output. The MCH should stay full and active until the MCH 100,000-microsecond timeout deactivate occurs.

If the Write FF did not get cleared by the function signal (a hardware fault), the MAC will accept the data byte and make the MCH empty.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0
a. Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
b. Send Write Reference ROM Memory Function to MAC; Op Code equals five, Type equals three.
c. Activate MCH.
d. Send CW1 and CW2, both equal zero.
e. Deactivate MCH.
f. Activate MCH.
g. Output four bytes of 00 data to Reference ROM.
h. Deactivate MCH.
i. Report Channel Error Status and error if proper MCH/MAC responses did not occur during any of the above steps a through h; expected result equals zero.

Conditions:

C.1
a. Send Master Clear function to MAC; Op Code equals six, Type equals zero.
b. Activate MCH.
c. Send two dummy Control Words (CW1 and CW2 equal zero).
d. Deactivate MCH.
e. Report Channel Error Status and error if proper MCH/MAC responses did not occur during any of the above steps a through d; expected result equals zero.

C.2
a. Activate MCH.
b. Output one data word of 3404 and record time until MCH goes empty, or 100,100 microseconds.
c. Report time until empty received in microseconds; error if time is less than 99990 or equals 100,100 microseconds (bytes 1, 2, and 3).

C.3 Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

Section 1, Subsection 9

This subsection tests the clearing of the Read FF by the function signal.

Method:

The basic method used to verify that the Read FF clears when MAC functions are sent is as follows:

- Perform a read function from the MAC Reference ROM memory. This operation sets the Read FF (FF is on the BZ ZIF board at C1E).
- Input one byte of data and then deactivate the MCH.
- Send a Master Clear function to MAC. The function signal should clear the Read FF.
- Activate the MCH, send two dummy Control Words to MAC, and then deactivate MCH.
- Activate the MCH and check that MAC does not send a word back. A properly operating MAC should not respond to the Activate. The MCH should stay empty and active until the MCH 100,000 microsecond timeout deactivate occurs.

  If the Read FF did not get cleared by the function signal (a hardware fault), the MAC will send a data byte and make the MCH full.

Initialization:

- Clear IOU Fault Status Register and clear MCH channel error if set.

Conditions:

C.0
a. Send Master Clear Function to MAC; Op Code equals six, Type equals zero.
b. Send Read Reference ROM Memory Function to MAC; Op Code equals four, type equals three.
c. Activate MCH.
d. Send CW1 and CW2, both equal to zero.
e. Deactivate MCH.
f. Activate MCH.
g. Input one byte of data from the Reference ROM.
h. Deactivate MCH.
i. Report Channel Error Status and error if proper MCH/MAC responses did not occur during any of the above steps a through h; expected result equals zero.

C.1
a. Send Master Clear Function to MAC; Op Code equals six, type equals zero.
b. Activate MCH.
c. Send two dummy Control Words (CW1 and CW2 equal zero).
d. Deactivate MCH.
e. Report Channel Error Status and error if proper MCH/MAC responses did not occur during any of the above steps a through d; expected result equals zero.

C.2
a. Activate MCH.
b. Report time until full received (if it is) or MCH goes inactive (bytes 1, 2, and 3); error if full received or deactivate time is less than 99990 or equals 100,100 microseconds.

C.3 Report Channel Error Status (error if MCH is not inactive and empty; expected result equals zero).

Section 01, Subsection 10

This subsection verifies that the PROM data can be read without error. Checks will be performed on the data read in an attempt to verify the validity of the data. The clock PROM's will be tested in reverse order of their position on the data bus. This should permit isolation of data path failures to two FRU's.

Code in this subsection will also attempt to read and save the current contents of the delay registers. This data will then be written back to the registers when the clock delay testing is complete.

Conditions

C.0 - Report the ID of the currently selected clock PROM. (information only).

C.1 - Read the data from the currently selected clock delay PROM. Report channel status in bits 60-63. See status format 1. Error if not zero.

Conditions:

C.2 - Check board identification information for a match with the clock select code. Report expected and actual data. Error if not equal.

Repeat subsection for next clock PROM.
End subsection

Section 01, Subsection 11

This subsection tests the MAC address bus and parity checkers applicable to the clock delay register system.

Method:
Each clock delay register is intialized with data zero. No error check is made during this operation. Then each delay register is selected in turn. Status checks are made after each selection. An error will cause the test to halt with the condition information described below.

Conditions

C.0 - Write all clock delay registers with zero data. Clear CPU errors. Read and report the clock parity status(format 2). Error if nonzero.

C.1 - Report the clock select to be tested. Information report.

C.2 - Select the clock register being tested. Select a nonexistant clock(id=1F). Read and report the clock status(format 2). Error if nonzero.

If not last clock select, Repeat from C.1 for the next clock select code.

End subsection

Section 01, Subsection 12

This subsection tests that the clock data registers can be written without error and that they will retain the data written. Each clock delay register is tested in turn.

Method:
Each clock delay register is initialized with data zero. No error check is made during this operation. Then each clock is written with a data pattern. The data is then read and verified against the data written. Parity status is also checked and an error report is generated if any of the clock delays has a parity error. The pattern of parity errors reported should provide the information necessary to isolate the failing module. Different data patterns will be used. The clock delay registers will be tested in order of their position on the data bus.

Conditions

C.0 - Initialize each clock delay register with zero data. Clear errors. Read and report clock parity status(format 2). Error if not zero.

C.1 - Report the clock register delay selected. Information report only.

C.2 - Write the current data pattern to the selected clock delay register. Collect and report channel status(format 1). Error if not zero.

C.3 - Read data from the clock register. Report channel status accummulated during the run(format 1). Error if nonzero.

Note - The read buffer is preset with the expected data so that the following data check is correct even though the full buffer is not read.

C.4 - Verify the data read against data written(the first 8 bytes). Report actual and expected values. Error if not equal.

C.5 - Verify the data read against data written(the second 8 bytes). Report actual and expected values. Error if not equal.

C.6 - Verify the data read against data written(the third 8 bytes). Report actual and expected values. Error if not equal.

C.7 - Verify the data read against data written(the fourth 8 bytes). Report actual and expected values. Error if not equal.

NOTE. status should include bits for each clock parity report, bit for errors detected on select, and channel errors.

Repeat from C.1 for each clock delay register.
Repeat from C.1 for each data pattern.
End subsection

Section 01, Subsection 13

This subsection verifies that the clock delay registers can all be written before they are read. It also verifies that the reading of one register does not disturb the data in another.

Method:

All clock delay registers are written with zero data. Then, in turn, each delay register is read, the data read is verified, and data ones are written back. Finally, each delay register is read and the data is compared against data ones.

Conditions

C.0 - Write all clock delay registers with zeros data. Check channel status after each write. Report select code of last register written. The report will contain -EE- in byte 0 of the report if there was a channel error at completion of the write. Error if any channel error detected.

C.1 - Read the data from each clock delay and write ones back to that register. Verify the data read against the data written in C.0. Report expected and actual values. Error if any difference.

C.2 - Report channel status read in previous condition. Error if nonzero. In byte 0, report the ID of the last clock delay read in the previous condition. Bytes 4,5 indicate status after selecting the clock. Bytes 6,7 indicate status after the read.

C.3 - Read and verify the data from each clock delay register. Report the expected and actual values. Error if any differences.

C.4 - Report channel status read in previous condition. Error if nonzero. In byte 0, report the ID of the last clock delay read in the previous condition.

End subsection.

Section 01, Subsection 14

This subsection tests the parity detection and reporting of the clock fanout arrays.

Method:
All the clock delay registers will be written with zero. Errors will then be cleared. Next a clock delay register will be written with a parity error. Clock parity status will then be read and verified to see that the parity error was detected and reported properly. This sequence will then be repeated for the other clock delay registers.

Conditions:

C.0 - Write all delay registers with zero data. Clear CPU errors. Read and report the contents of the clock status registers. Error if not zero.

C.1 - Report the clock register currently selected for test. Information only.

C.2 - Write clock delay with a parity error in one of the clock delay registers on the selected array. Report maintenance channel status. Error if not zero. (See Status report format 1)

Conditions:

C.3 - Clear CPU errors. Read and report the clock parity status. Error if not as expected. (See Status report format 2)

Repeat from C.2 for all arrays.

Section 01, Subsection 15

The purpose of this subsection is to test the detection, by the clock fanout arrays, of addressing parity errors.

Method:
Write all clock delay registers with zero. Then, clear CPU errors. Issue clock selects with parity errors. Read and verify the clock parity status after each select attempt. Each of the clock fanout arrays should report an error. Repeat for all the possible clock select codes.

C.0 - Write all delay registers with zero data. Clear CPU errors. Read and report the clock parity status. Error if not zero. (See status report format 2)

C.1 - Report the currently selected clock register. Information only.

C.2 - Attempt to select a clock array register using a select code with a parity error. Report channel status. Error if not zero. (See status report format 1)

C.3 - Read and report the clock parity status. Error if not as expected. (See status report format 2) Note - since all of the parity checkers are looking at the same address they should all report the same.

Repeat subsection for each clock array.
END OF SUBSECTION.

SECTION 2 - TEST MAC DATA BUS PATHS

This section verifies that data can be passed the entire length of the MAC Data Bus path from the MAC ZIF boards back to the MAC ZIF boards without error. The stub legs off of the main MAC Data Bus path are also verified for passing data correctly.

Subsection 0 - This subsection tests the MAC Data Bus path from the PFS Register Boards to MCH for passing zero data, and tests that certain PFS Register bits are clear.

Subsection 1 - This subsection tests the MAC Data Bus path from the MAC BK ZIF board back to the MAC BK ZIF board.

Subsection 2 - This subsection tests the MAC Data Bus path from AC Soft Control 2 Auxiliary Board through the ALN Soft Control Auxiliary Board to the IF Decode RAM auxiliary Boards.

Subsection 3 - This subsection tests the MAC Data Bus path legs from the Register File Auxiliary Board to the BDP Port A and B Auxiliary Boards, and from AC Soft Control 1 Auxiliary Board to the BDP RAM PAK 0 and RAM PAK 1 Auxiliary Board.

Hardware Tested:

- The MAC Data Bus circuitry on the MAC BK ZIF board and the related paths to and from the MAC Data Bus on the MAC BZ ZIF board.
- The MAC Data Bus path through the AC Soft Control Memory Auxiliary Boards, the Register File Auxiliary Boards, the PFS Register Boards and the Control Store Disassembly Network LSI Arrays.
- The MAC Data Bus path through the BDP Port A and B Auxiliary Boards and the BDP RAM PAK 0 and RAM PAK 1 Auxiliary Boards.
- The MAC Data Bus path through the ALN Soft Control Memory Auxiliary Board and the IF Decode RAM Auxiliary boards and LSI arrays.
- A partial check of the PFS Register Boards, including the DEC and PTM Registers.
- A partial check of the data parity bit from the MAC BZ ZIF board to the IOU MCH.

Method:

Refer to the Method: documentation provided with each subsection.

Section 2, Subsection 0

This subsection tests the MAC Data Bus path from the PFS Register Boards to MCH for passing zero data, and tests that certain PFS Register bits are clear.

Method:

The subsection attempts to ensure that certain bits in the PFS Registers are clear and that these zero bits can be read by the MCH without error.

All bits within the PFS Registers cannot be cleared because hardware design requires all processor hardware to be exercised after a power up in order to ensure good data throughout the processor. At this point in the testing of the processor, we cannot exercise all areas of the processor, thus many parity errors are being continually clocked into the PFS Registers. The only PFS Register bits that should be clear are those for hardware exercised in sections 0 and 1 (MAC hardware, Soft Control Memories, Control Store Memory and Register File).

This subsection will read the contents of each PFS Register and check the data bits which should be zero for being zero.

In addition, the subsection will check to see if a MCH channel error occurred after each byte read. If a MCH channel error occurs, it means that either a PFS Register Board Parity Generator is bad, or a bit has dropped or picked in the read path, or the MAC output Register parity checker is bad.

Initialization:

- Clear IOU Fault Status Register and clear MCH channel error if set.

Repeat the following conditions for each PFS Register (89 (hex) through 80 (hex).

Conditions:

C.0
a. Master Clear processor.
b. Clear Errors in processor.
c. Report PFS Register being read; informational.

C.1 Report bits of PFS Register that will be checked to be 0; informational. A zero bit in a bit position means check that bit position from the PFS Register for being clear, a one bit means does not care.

C.2 Read PFS Register xx.
a. Initiate MAC Read Function.
b. Set input byte count equal to zero.
c. Input one byte from MAC and save.
d. Check MCH channel error and if first error this input save input byte count.
e. Increment input byte count (byte count plus one).
f. Repeat steps c through e if byte count not equal to eight.
g. Deactivate MCH.
h. Report actual data read from PFS Register; error if all checked bits are not clear.

Conditions:

C.3 If MCH channel error occurred in C.2 report byte count on which first channel error was sensed and report error; expected result equals zero. Byte count reported in Condition Result byte 7.

C.4 If MCH channel error occurred in C.2 report IOU Fault Status Register 2 bytes 5, 6 and 7 and report error; expected result equals zero.

If Repeat Iteration set, go to C.2.

Exit subsection if end of PFS Registers.

PFS Register Operands:

All values are in Hexadecimal.

Order of test	PFS Number	Bits to be checked in PFS Reg for being clear+
1	89	FFFF FF00 0000 0000
2	88	FFFF FFFF FFFF FF7C
3	87	FFFF FF00 0000 0000
4	86	0000 0003 011F 1EFE
5	85	FFFA FF00 0000 0000
6	84	FFFF FFFF FFFF FF90
7	83	FFFF FF00 0000 0000
8	82	FFFF FFFF FFF8 F8FE
9	81	C30F FF00 0000 0000
10	80	FFFF FFFF FFF1 F8FE

+ Bit positions containing zero bits should also be clear in the corresponding PFS Register bit positions. Bit positions containing one bits are don't care bit positions; that is, the corresponding bit positions in the PFS Register can be set or clear.

Isolation Notes:

The order of testing the PFS Registers is such that the PFS Register board closest to MACs Data Mux 5 is tested first, followed by the next farther board, etc.

Testing of PFS Register 89 only checks that the zero fill gate on MACs Data Mux 5 works; thus if an error occurs in bytes 2 through 7 of PFS Register 89 data, the problem should be in one of the MAC boards (BZ or BK).

If a bit is set on PFS Register data when it should be clear, caution should be used when trying to determine where the faulty hardware really exists. A PFS Register bit can be set for any one of the following reasons:

- The PFS Register board is faulty.

- The MAC Data Bus is faulty somewhere from the PFS Register to MAC Data Mux 5.

- A parity checker or its error line to the PFS Register is faulty.

- The data being checked by the parity checker is bad. This could occur for numerous reasons; data of zeros was written in section 1 of this test; the write path may be faulty or the actual memory or register may be faulty.

Section 2, Subsection 1

This subsection tests the MAC Data Bus path from the MAC BK ZIF board back to the MAC BK ZIF board.

Method:

The MAC Data Bus path comprises the two MAC ZIF boards, 10 Auxiliary Boards, and 10 LSI Arrays. These subassemblies occur in the following order: MAC BZ ZIF board, MAC BK ZIF board, AC SC2 Auxiliary PAK, AC SC1 Auxiliary PAK, Register File Auxiliary PAK at B3A1, PFS Auxiliary Board 0, PFS Auxiliary Board 1, PFS Auxiliary Board 2, PFS Auxiliary Board 3, PFS Auxiliary Board 4, Control Store Dissasembly Register LSI Arrays, MAC BK ZIF board, MAC BZ ZIF board.

In order to isolate any failure in the MAC Data Bus path to three or less subassemblies, the parity checkers on each auxiliary board must be employed. To accomplish this, each AC SCM is written with the test pattern (location 0), as is Register File location 0 and the DEC and PTM registers on the PFS boards. This writing sends data along the MAC Data Bus path.

Location 0 of the AC Soft Control memories are then swept so the data that was written is parity checked. The Register File location 0 data is read so the data written is parity checked. The DEC and PTM registers data is read and checked for proper data as well as being parity checked as it flows along the MAC Data Bus path. The results from all of these parity checks are accumulated, formatted, and reported.

If any hardware fault exists along the MAC Data Bus path it will be indicated by bad data being read from the DEC or PTM register. The parity checker information is then used to determine where along the BUS Data path the fault exists (data written into the DEC and PTM registers drops or picks a bit prior to reaching the registers, or the data drops or picks a bit after leaving the registers when being read.

NOTE

Bit 18 of PTM is never set in this subsection because bit 18 of PTM forces bad parity in the MAC logic.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Master Clear processor.
b. Write AC Soft Control Memory Boards with first four bytes of Pattern, location 0 each board; address 40 (hex) (AC SC2) and address 80 (hex) (AC SC1).
c. Write location 0 of the register file with Pattern.
d. Write the DEC Register with Pattern.
e. Write the PTM Register with Pattern (bit 18 cleared if set in pattern).
f. Clear Errors in processor.
g. Report the Pattern Number in Condition Result byte 7; informational.

C.1
a. Read (sweep) location 0 from AC SC2 (address 40 (hex).
b. Read (sweep) location 0 from AC SC1 (address 80 (hex).
c. Read Register File location 0 ignoring the data read.
d. Report the Pattern data being used; informational.
e. Clear processor PFS Register 86 (hex).

C.2 Read DEC Register data and check.
a. Report data read; error if bits 32 through 63 do not equal bits 32 through 63 of Pattern and bits 0 through 31 do not equal zeros. See Parity Error Status Report following.

C.3 Read PTM Register data and check.
a. Report data read; error if bits 0 through 47 do not equal bits 0 through 47 of Pattern and bits 48 through 63 do not equal zeros. Bit 18 of data should always be clear.

C.4 Collect all relevant parity error status from PFS Registers and report; error if not zero.

Exit subsection if end of test patterns.

Patterns (Hexadecimal):

0	0000 0000 0000 0000	(byte parity bits equal zero)
1	FFFF FFFF FFFF FFFF	(byte parity bits equal zero)
2	5555 5555 5555 5555	(byte parity bits equal zero)
3	AAAA AAAA AAAA AAAA	(byte parity bits equal zero)
4	DFDF DFDF DFDF DFDF	(byte parity bits equal one)
5	DF00 DF00 DF00 DF00	(even byte parity bits equal one, odd bytes equal zero)
6	00DF 00DF 00DF 00DF	(even byte parity bits equal zero, odd bytes equal one)
7	0000 0000 0000 0000	(byte parity bits equal zero)

Parity Error Status Report (Condition 4):

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	AC SC2 (loc A2E1)	80	45
62	AC SC1 (loc A2G1)	80	46
61	Reg File Board A (loc B3A1)	84	57
60	Reg File Board B (loc B3A2)	84	58
59	PFS Board 0 (loc A2E2)	80	63
58	PFS Board 1 (loc A2G2)	82	63
57	PFS Board 2 (loc B2C2)	84	63
56	PFS Board 3 (loc C2B2)	86	63
55	PFS Board 4 (loc C2K2)	88	63
54	MAC Data Mux 5 (loc C1D)	86	35
53	MAC Chan Out Reg (loc C1E)	86	32

Section 2, Subsection 2

This subsection tests the MAC Data Bus path leg from AC Soft Control 1 Auxiliary Board through the ALN Soft Control Auxiliary Board to the IF Decode RAM Auxiliary Boards.

Method:

The only way to determine whether or not a fault exists in this leg of the MAC Data Bus path is via the parity checkers on the Soft Control Memory read data paths.

The method used in this subsection is similar to that used in Section 2, Subsection 1. A pattern is written to location 0 of the RAMs in each Auxiliary Board (AC SC1, ALN SC, IF C180 Decode RAMs on IF Decode RAM Boards 0 and 1, IF C170 Decode RAMs on IF Decode RAM Boards 0 and 1). Location 0 of each of these Soft Control Memories is swept and the data is parity checked. The results of these parity checks are accumulated, formatted and reported.

If any hardware fault exists along this leg of the MAC Data Bus path, the parity checker information is used to isolate the failure to the faulty board.

Repeat the following conditions for each test pattern.

Conditions:

C.0
- a. Master Clear processor.
- b. Write AC Soft Control Memory 1, location 0 (address 80 (hex), with Pattern.
- c. Write ALN Soft Control Memory, location 0 (address C0 (hex), with Pattern.
- d. Write IF C180 Decode Memory, location 0 (CW1 equals zero, CW2 equals zero), with Pattern.
- e. Write IF C170 Decode Memory, location 0 (CW1 equals one, CW2 equals zero), with Pattern.
- f. Clear Errors in processor.
- g. Report the Pattern Number in Condition Result byte 7; informational.

C.1
- a. Read (sweep) AC Soft Control Memory 1, location 0 (address 80 (hex).
- b. Read (sweep) ALN Soft Control Memory, location 0 (address C0 (hex).
- c. Read (sweep) IF C180 Decode Memory, location 0 (CW1 equals zero, CW2 equals zero).
- d. Read (sweep) IF C170 Decode Memory, location 0 (CW1 equals one, CW2 equals zero).
- e. Report the Pattern data being used; informational.

C.2 Collect all relevant parity error status from the PFS Register and report; error if not zero.

Exit subsection if end of test patterns.

Patterns (Hexadecimal):

0	0000 0000 0000 0000	(byte parity bits equal zero)
1	FFFF FFFF FFFF FFFF	(byte parity bits equal zero)
2	5555 5555 5555 5555	(byte parity bits equal zero)
3	AAAA AAAA AAAA AAAA	(byte parity bits equal zero)
4	DFDF DFDF DFDF DFDF	(byte parity bits equal one)
5	DF00 DF00 DF00 DF00	(even byte parity bits equal one, odd bytes equal zero)
6	00DF 00DF 00DF 00DF	(even byte parity bits equal zero, odd bytes equal one)
7	0000 0000 0000 0000	(byte parity bits equal zero)

NOTE

Bits 0,1 8,9 16,17 24,25 are cleared when pattern is written into C170 Decode RAM. Bits 0,1,2 8,9,10 16,17,18 24,25,26 are cleared when pattern is written into C180 Decode RAM.

Parity Error Status Report (Condition 2):

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	AC SCM 1(loc A2G1)	80	46
62	ALN SCM (loc C2K1)	80	44
61	IF PAK 0 C180 Decode (loc C2A1)	86	42
60	IF PAK 0 C170 Even Decode (loc C2A1)	86	41
59	IF PAK 0 C170 Odd Decode (loc C2A1)	86	40
58	IF PAK 1 C180 Decode (loc C2B1)	86	50
57	IF PAK 1 C170 Even Decode (loc C2B1)	86	49
56	IF PAK 1 C170 Odd Decode (loc C2B1)	86	48

Section 2, Subsection 3

This subsection tests the MAC Data Bus path legs from Register File Auxiliary Board to the BDP Port A and B RAM Auxiliary Boards, and from AC SC2 Auxiliary Board to the BDP RAM PAK 0 and RAM PAK 1 Auxiliary Boards.

Method:

The only way to determine whether or not a fault exists in either of these legs of the MAC Data Bus paths is via the parity checkers on the RAM read data paths.

The method used in this subsection is similar to that used in Section 2, Subsection 1. A pattern is written to location 0 of the RAMs in each Auxiliary Board (Register File; AC SC2; BDP Port A Board; BDP Port B Board; BDP RAM PAK 0, EBCDIC Memory; BDP RAM PAK 1, Spec. Error Memory; BDP RAM PAK 0, Convert Memory; and BDP RAM PAK 1, X256 Memory). Each location 0 of these RAMs is swept and the data is parity checked. The results of these parity checks are accumulated, formatted and reported.

If any hardware fault exists along either of these legs of the MAC Data Bus paths, the parity checker information must be used to isolate the failure to the faulty board.

Repeat the following conditions for each test pattern.

Conditions:

C.0
- a. Master Clear processor.
- b. Write Register File location 0 with Pattern.
- c. Write AC SC2, location 0 (address 40 (hex), with Pattern.
- d. Write BDP Port A RAM, location 0 (CW1 equals zero, CW2 equals zero), with Pattern.
- e. Write BDP Port B RAM, location 0 (CW1 equals one, CW2 equals zero), with Pattern.
- f. Write BDP RAM PAK 0, EBCDIC Memory, location 0 (CW1 equals 2, CW2 equals zero) with Pattern.
- g. Write BDP RAM PAK 0, Convert Memory, location 0 (CW1 equals three, CW2 equals zero) with Pattern.
- h. Write BDP RAM PAK 1, Spec. Error Memory, location 0 (CW1 equals four, CW2 equals zero) with Pattern.
- i. Write BDP RAM PAK 1, X256 Memory, location 0 (CW1 equals five, CW2 equals zero) with Pattern.
- j. Clear Errors in processor.
- k. Report the Pattern Number in Condition Result byte 7; informational.

C.1
- a. Read Register File location 0.
- b. Read (sweep) BDP Port A RAM, location 0 (CW1 equals zero, CW2 equals zero).
- c. Read (sweep) BDP Port B RAM, location 0 (CW1 equals one, CW2 equals zero).
- d. Report the Pattern data being used; informational.

C.2 Collect all relevant parity error status from the PFS Registers for Register File to the BDP Ports A and B RAMs leg of the MAC Data Bus paths, and report; error if not zero.

C.3
- a. Read (sweep) AC SC2, location 0 (address 40 (hex).
- b. Read (sweep) BDP RAM PAK 0, EBCDIC Memory, location 0 (CW1 equals two, CW2 equals zero).
- c. Read (sweep) BDP RAM PAK 0, Convert Memory, location 0 (CW1 equals three, CW2 equals zero).
- d. Read (sweep) BDP RAM PAK 1, Spec. Error Memory, location 0 (CW1 equals four, CW2 equals zero).
- e. Read (sweep) BDP RAM PAK 1, X256 Memory, location 0 (CW1 equals five, CW2 equals zero).
- f. Collect all relevant parity error status from the PFS Registers for AC SC2 to the BDP RAM PAK 0 and PAK 1 leg of the MAC Data Bus paths, and report; error if not zero.

If Repeat Iteration set, go to C.1.

Exit subsection if end of test patterns.

Patterns (Hexadecimal):

0	0000 0000 0000 0000	(byte parity bits equal zero)
1	FFFF FFFF FFFF FFFF	(byte parity bits equal zero)
2	5555 5555 5555 5555	(byte parity bits equal zero)
3	AAAA AAAA AAAA AAAA	(byte parity bits equal zero)
4	DFDF DFDF DFDF DFDF	(byte parity bits equal one)
5	DF00 DF00 DF00 DF00	(even byte parity bits equal one, odd bytes equal zero)
6.	00DF 00DF 00DF 00DF	(even byte parity bits equal zero, odd bytes equal one)
7.	0000 0000 0000 0000	(byte parity bits equal zero)

Parity Error Status Report (for Condition 2):

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	Reg File Board A (loc B3A1)	84	57
62	Reg File Board B (loc B3A2)	84	58
61	BDP Port A PAK (loc A3B1)	81	4
61	BDP Port B PAK (loc A3C1)	81	5

Parity Error Status Report (for Condition 3):

63	AC SC2 (loc A2E1)	80	45
62	BDP RAM PAK 0, EBCDIC (loc A3A2)	81	9
61	BDP RAM PAK 0, Convert (loc A3A2)	81	11
60	BDP RAM PAK 1 (loc A3B2)	81	3

SECTION 3 - TEST PFS REGISTER BOARDS

This section completes the verification of the PFS Register Boards logic and the MAC ZIF Boards control of the PFS Register Boards logic, as much as possible this early in the processor testing sequence.

Subsection 0 - This subsection tests the byte and board select signals sent to and used on the PFS Register Boards, as well as related control logic on the MAC ZIF boards.

Subsection 1 - This subsection tests the setting of all bits in the PFS Registers by writing the PFS Registers with byte data with bit 7 set to a one (causes all PFS Register bits to set).

Subsection 2 - This subsection tests the clearing of PFS Register bits by writing the PFS Registers with byte data with bit 7 clear. All bits cannot be guaranteed to clear at this point in the processor testing sequence; a partial test.

Subsection 3 - This subsection tests the PFS Board Select Translation logic on the MAC BK ZIF board, and the Error Log Board Select signals sent to the PFS Register Auxiliary Boards.

Subsection 4 - This subsection tests that a Clear Errors Function clears the PFS Register bits; a partial test.

Hardware Tested:

- The MAC BK and BZ ZIF boards; logic portions used to control the PFS Register Auxiliary boards.
- The PFS Register Auxiliary boards PFS Register data and control logic.

Method:

Refer to the Method: documentation provided with each subsection for detailed information.

It should be noted here that the hardware clearing of PFS Register bits, by writing zeros and by Clear Errors Function, cannot be completely verified at this point in the processor testing sequence; all PFS Register bits may not be clearable. Only those bits that should be clearable are verified as being clearable in this test (MCT3/P). The other PFS Register bits will be verified as clearable bit by bit as later tests verify operation of functional areas of processor connected to these bits.

Section 3, Subsection 0

This subsection tests the byte and board select signals sent to and used on the PFS Register Boards, as well as related control logic on the MAC ZIF boards.

Method:

The subsection writes a 64-bit byte pattern to the DEC and PTM Registers, which reside on the PFS Register Boards. The DEC and PTM Registers data is then read and verified for correctness. In addition, the status of the parity checkers in the DEC and PTM read path is collected, formatted and checked to ensure that no parity errors occurred during the read.

Any hardware faults related to byte and board select signals generated on the MAC ZIF boards and sent to the PFS Register Boards should be indicated by the wrong byte of data being read (or written) from the DEC and/or PTM Registers.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Master Clear processor.
b. Clear Errors in processor.
c. Write the DEC Register (30 (hex) with test pattern.
d. Report the Pattern Number; informational.

C.1
a. Write the PTM Register (A0 (hex) with test pattern.
b. Report the pattern data; informational.

C.2 Read the DEC Register data.
a. Report the data read; error if bits 32 through 63 do not equal bits 32 through 63 of test pattern and bits 0 through 31 do not equal zeros.

C.3 Read the PTM Register data.
a. Report the data read; error if bits 0 through 47 do not equal bits 0 through 47 of test pattern and bits 48 through 63 do not equal zeros.

C.4 Collect all relevant parity error status from the PFS Registers and report; error if not zero.

Conditions:

C.5 Check for no channel errors in MCH.
a. Report MCH channel error by storing IOU Fault Status Register 2 bytes 5, 6 and 7 in Condition Result; right justified; expected result equals zero.

Exit subsection if end of test patterns.

Patterns (Hexadecimal):

Bit 18 must always be zero.

0	0819	0A3B	4C5D	6E7F	
1	F7E6	D5C4	B3A2	9180	
2	1011	1011	1011	1011	(check parity bits, alternate byte parity)
3	1101	1101	1101	1101	(check parity bits, alternate byte parity)
4	0000	0000	0000	0000	

Parity Error Status Report (condition 4):

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	PFS Board 0 (loc A2E2)	80	63
62	PFS Board 1 (loc A2G2)	82	63
61	PFS Board 2 (loc B2C2)	84	63
60	PFS Board 3 (loc C2B2)	86	63
59	PFS Board 4 (loc C2K2)	88	63
58	MAC Data Mux 5 (loc C1D)	86	35
57	MAC Chan Out Reg (loc C1E)	86	32

Section 3, Subsection 1

This subsection tests the setting of all bits in the PFS Registers by writing the PFS Registers with byte data with bit 7 set to a one (causes all PFS Register bits to set).

Method:

The subsection writes data to each PFS Register (80 (hex) through 89 (hex) which should cause the PFS board control to set all bits in the PFS Register (byte data with bit 7 set to a one).

The subsection then reads each PFS Register and verifies that the data has all bits set. It also checks to see if a MCH channel error occurs while reading the data, and reports the number of the first condition that senses the channel error.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0 a. Master Clear processor
b. Write each PFS Register (80 (hex) through 89 (hex) with a 0101 0101 0101 0101 (hex) pattern (sets all bits).
c. No valid data to report from this condition, so report 0-0.

C.1 Read PFS Register 80 (hex).
a. Report Data Read (error if not equal to FFFF FFFF FFFF FFFF (hex) or MCH channel error occurred).

C.2 Read PFS Register 81 (hex).
a. Report Data Read; error if not equal to FFFF FF00 0000 0000 (hex) or MCH channel error occurred.

C.3 Read PFS Register 82 (hex).
a. Report Data Read; error if not equal to FFFF FFFF FFFF FFFF (hex) or MCH channel error occurred.

C.4 Read PFS Register 83 (hex).
a. Report Data Read; error if not equal to FFFF FF00 0000 0000 (hex) or MCH channel error occurred.

C.5 Read PFS Register 84 (hex).
a. Report Data Read; error if not equal to FFFF FFFF FFFF FFFF (hex) or MCH channel error occurred.

C.6 Read PFS Register 85 (hex).
a. Report Data Read; error if not equal to FFFF FF00 0000 0000 (hex) or MCH channel error occurred.

C.7 Read PFS Register 86 (hex).
a. Report Data Read; error if not equal to FFFF FFFF FFFF FFFF (hex) or MCH channel error occurred.

C.8 Read PFS Register 87 (hex).
a. Report Data Read; error if not equal to FFFF FF00 0000 0000 (hex) or MCH channel error occurred.

C.9 Read PFS Register 88 (hex).
a. Report Data Read; error if not equal to FFFF FFFF FFFF FFFF (hex) or MCH channel error occurred.

C.10 Read PFS Register 89 (hex).
a. Report Data Read; error if not equal to FFFF FF00 0000 0000 (hex) or MCH channel error occurred.

Conditions:

C.11 If MCH channel error occurred in C.1 through C.10; report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition that sensed a channel error in condition result byte 4; expected result equals zero.

Section 3, Subsection 2

This subsection tests the clearing of PFS Register bits by writing the PFS Registers with byte data with bit 7 clear. All bits cannot be guaranteed to clear at this point in the processor testing sequence; a partial test.

Method:

The subsection writes data to each PFS Register (80 (hex) through 89 (hex) which should cause the PFS board control to clear all bits in the PFS Register (byte data with bit 7 clear). However, because the entire processor has not been exercised at this point in the processor testing, bad data exists in portions of processor which will be causing parity errors which in turn cause bits in the PFS Registers to be continually set. Therefore, all bits in the PFS Registers will not be able to be cleared. Which bits will be set is unknown. They vary from one Power On to the next Power On.

The subsection then reads each PFS Register and verifies that the data bits which should be clearable (hardware exercised by previous sections) are clear. The other bits, those which may or may not be clear, are ignored. The subsection also checks to see if a MCH channel error occurs while reading the data, and reports the number of the first condition that senses the channel error.

NOTE

The expected results for each condition below is expressed in the form of "error if not equal to *** FFFF FFFF FFF8 FEFE (HEX), where a one bit means don't care, the bit can be a one or a zero in the data read, and a zero bit means that the corresponding bit in the data read must also be zero.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set. Also, set all PFS Registers (80 (hex) through 89 (hex) to F through F (hex).

Conditions:

C.0
a. Master Clear processor.
b. Write each PFS Register (80 (hex) through 89 (hex) with a FEFE FEFE FEFE FEFE (hex) pattern (attempts to clear all bits).
c. No valid data to report from this condition, so report 0-0.

C.1 Read PFS Register 80 (hex).
a. Report Data Read; error if not equal to *** FFFF FFFF FFF1 F8FE (hex) or MCH channel error occurred.

C.2 Read PFS Register 81 (hex).

a. Report Data Read; error if not equal to *** C30F 0000 0000 0000 (hex) or MCH channel error occurred.

C.3 Read PFS Register 82 (hex).
a. Report Data Read; error if not equal to *** FFFF FFFF FFF8 F8FE (hex) or MCH channel error occurred.

C.4 Read PFS Register 83 (hex).
a. Report Data Read; error if not equal to *** FFFF FF00 0000 0000 (hex) or MCH channel error occurred.

C.5 Read PFS Register 84 (hex).
a. Report Data Read; error if not equal to *** FFFF FFFF FFFF FF90 (hex) or MCH channel error occurred.

C.6 Read PFS Register 85 (hex).
a. Report Data Read; error if not equal to *** FFFA FF00 0000 0000 (hex) or MCH channel error occurred.

C.7 Read PFS Register 86 (hex).
a. Report Data Read; error if not equal to *** 0000 0003 011F 1EFE (hex) or MCH channel error occurred.

C.8 Read PFS Register 87 (hex).
a. Report Data Read; error if not equal to *** FFFF FF00 0000 0000 (hex) or MCH channel error occurred.

C.9 Read PFS Register 88 (hex).
a. Report Data Read; error if not equal to *** FFFF FFFF FFFF FF7C (hex) or MCH channel error occurred.

C.10 Read PFS Register 89 (hex).
a. Report Data Read; error if not equal to *** FFFF FF00 0000 0000 (hex) or MCH channel error occurred.

C.11 If MCH channel error occurred in C.1 through C.10; report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition that sensed the channel error in Condition Result byte 4; expected result equals zero.

Section 3, Subsection 3

This subsection tests the PFS Board Select Translation logic on the MAC BK ZIF board, and the PFS Board Select signals sent to the PFS Register Auxiliary Boards.

Method:

The subsection must use an addressing test method in order to determine that the PFS Board Select Translation hardware functions correctly.

All PFS Registers are set to F--F (hex). Then one PFS Register is written with zeros (starting with 80 (hex). All PFS Registers are read and only the register that was written with zeros should contain zeros. All of the other PFS Registers should still contain F--F (hex). This sequence is then repeated by writing zeros to the next PFS Register, and so on.

The DEC and PTM registers are initially written with F--F (hex), and are read each iteration of the test to verify that their contents have not been altered. The DEC and PTM registers being set to F--F (hex) also provides some sensitization to Mux C of the PFS Register Boards.

NOTE

The expected results for a cleared PFS Register are expressed in the form of "error if not equal to *** FFFF FFFF FFF8 FEFE (hex), where a one bit means don't care, the bit can be a one or a zero in the data read, and a zero bit means that the corresponding bit in the data read must also be zero.

Initialization:

- Master Clear processor.
- Write DEC Register with 0000 0000 FFFF FFFF (hex).
- Write PTM Register with FFFF DFFF FFFF 0000 (hex). (Bit 18 must be clear, otherwise causes MAC to force bad parity).
- Set all PFS Registers (80 (hex) through 89 (hex) to F--F (hex).
- Clear IOU Fault Status Registers.
- Clear MCH channel error if set.
- Set N equal to 80 (hex).
- Write PFS Register 80 (hex) with 0--0.

Repeat the following conditions for each value of N from N equals 80 (hex) through N equals 89 (hex), and N equals 30 (hex) and A0 (hex).

Conditions:

C.0 Report value of N in Condition Result byte 7; informational.

C.1 Read PFS Register 80 (hex).
a. Report data read.
b. If N equals 80 (hex), report error if data not equal to *** FFFF FFFF FFF1 F8FE (hex). Else, report error if data not equal to FFFF FFFF FFFF FFFF (hex), or MCH channel error occurred.

C.2 Read PFS Register 81 (hex).
a. Report data read.
b. If N equals 81 (hex), report error if data not equal to *** C30F 0000 0000 0000 (hex). Else, report error if data not equal to FFFF FF00 0000 0000 (hex), or MCH channel error occurred.

C.3 Read PFS Register 82 (hex).
a. Report data read.
b. If N equals 82 (hex), report error if data not equal to *** FFFF FFFF FFF8 F8FE (hex). Else, report error if data not equal to FFFF FFFF FFFF FFFF (hex), or MCH channel error occurred.

C.4 Read PFS Register 83 (hex).
a. Report data read.
b. If N equals 83 (hex), report error if data not equal to *** FFFF 0000 0000 0000 (hex). Else, report error if data not equal to FFFF FF00 0000 0000 (hex), or MCH channel error occurred.

C.5 Read PFS Register 84 (hex).
a. Report data read.
b. If N equals 84 (hex), report error if data not equal to *** FFFF FFFF FFFF FF90 (hex). Else, report error if data not equal to FFFF FFFF FFFF FFFF (hex), or MCH channel error occurred.

C.6 Read PFS Register 85 (hex).
a. Report data read.
b. If N equals 85 (hex), report error if data not equal to *** FFFA 0000 0000 0000 (hex). Else, report error if data not equal to FFFF FF00 0000 0000 (hex), or MCH channel error occurred.

C.7 Read PFS Register 86 (hex).
a. Report data read.
b. If N equals 86 (hex), report error if data not equal to *** 0000 0003 011F 1EFE (hex). Else, report error if data not equal to FFFF FFFF FFFF FFFF (hex), or MCH channel error occurred.

C.8 Read PFS Register 87 (hex).
a. Report data read.
b. If N equals 87 (hex), report error if data not equal to *** FFFF 0000 0000 0000 (hex). Else, report error if data not equal to FFFF FF00 0000 0000 (hex), or MCH channel error occurred.

Conditions:

C.9 Read PFS Register 88 (hex).
a. Report data read.
b. If N equals 88 (hex), report error if data not equal to *** FFFF FFFF FFFF FF7C (hex). Else, report error if data not equal to FFFF FFFF FFFF FFFF (hex), or MCH channel error occurred.

C.10 Read PFS Register 89 (hex).
a. Report data read.
b. If N equals 89 (hex), report error if data not equal to *** FFFF 0000 0000 0000 (hex). Else, report error if data not equal to FFFF FF00 0000 0000 (hex), or MCH channel error occurred.

C.11 Read DEC Register 30 (hex).
a. Report data read.
b. If N equals 30 (hex), report error if data not equal to 0--0. Else, report error if data not equal to 0000 0000 FFFF FFFF (hex), or MCH channel error occurred.

C.12 Read PTM Register A0 (hex).
a. Report data read.
b. If N equals A0 (hex), report error if data not equal to 0--0. Else, report error if data not equal to FFFF DFFF FFFF 0000 (hex), or MCH channel error occurred.

C.13 If MCH channel error occurred in C.1 through C.12; report error and contents of IOU Fault Status Register 2 bytes 5, 6 and 7; and number of first condition that sensed the channel error in Condition Result byte 4; expected result equals zero.

- N not equal to 89 (hex), 30 (hex) nor A0 (hex).
a. Write PFS Register N with F--F (hex).
b. Increment N (N+1).
c. Write PFS Register N with 0--0.
d. Go to Condition 0 to repeat conditions with new N.

- N equals 89 (hex).
a. Write PFS Register 89 (hex) with F--F (hex).
b. Set N equal to 30 (hex).
c. Write DEC Register (30 (hex) with 0--0.
d. Go to Condition 0 to repeat conditions with new N.

- N equals 30 (hex).
a. Write DEC Register (30 (hex) with 0000 0000 FFFF FFFF (hex)
b. Set N equal to A0 (hex).
c. Write PTM Register (A0 (hex) with 0--0.
d. Go to Condition 0 to repeat conditions with new N.

- N equals A0 (hex).
a. Write DEC Register (30 (hex) with 0--0.

Section 3, Subsection 4

This subsection tests that a Clear Error Function clears the PFS Register bits in a partial test.

Method:

The subsection sets all bits in the PFS Registers 80 (hex) through 89 (hex). A Clear Error Function is then sent to processor. The PFS Registers are then read and each register's data is checked to verify that its bits have been cleared. Again note, because the entire processor has not been exercised at this point in the processor testing, some PFS Register bits will not be able to be cleared because parity errors are being sensed. Those bits that should be clearable by the Clear Error Function will be checked, the other bits will be ignored.

NOTE

The expected results for each condition below are expressed in the form of "error if not equal to *** FFFF FFFF FFF8 FEFE (hex)", where a one bit means don't care, the bit can be a one or a zero in the data read, and a zero bit means that the corresponding bit in the data read must also be zero.

Initialization:

- Send Master Clear Function to processor.
- Set all PFS Registers (80 (hex) through 89 (hex) to F--F (hex).

Conditions:

C.0 Send Clear Error Function to processor.
a. Nothing valid to report, so report 0--0.

C.1 Read PFS Register 80 (hex).
a. Report data read; error if not equal to *** FFFF FFFF FFF1 F8FE (hex).

C.2 Read PFS Register 81 (hex).
a. Report data read; error if not equal to *** C30F 0000 0000 0000 (hex).

C.3 Read PFS Register 82 (hex).
a. Report data read; error if not equal to *** FFFF FFFF FFF8 F8FE (hex).

Conditions:

C.4 Read PFS Register 83 (hex).
a. Report data read; error if not equal to *** FFFF FF00 0000 0000 (hex).

C.5 Read PFS Register 84 (hex).
a. Report data read; error if not equal to *** FFFF FFFF FFFF FF90 (hex).

C.6 Read PFS Register 85 (hex).
a. Report data read; error if not equal to *** FFFA FF00 0000 0000 (hex).

C.7 Read PFS Register 86 (hex).
a. Report data read; error if not equal to *** 0000 0003 011F 1EFE (hex).

C.8 Read PFS Register 87 (hex).
a. Report data read; error if not equal to *** FFFF FF00 0000 0000 (hex).

C.9 Read PFS Register 88 (hex).
a. Report data read; error if not equal to *** FFFF FFFF FFFF FF7C (hex).

C.10 Read PFS Register 89 (hex).
a. Report data read; error if not equal to *** FFFF FF00 0000 0000 (hex).

SECTION 4 - TEST MAC HARDWARE ERROR REPORTING

This section verifies that the parity checkers on the MAC ZIF Boards and the PFS Register Auxiliary Boards report parity errors, and that the proper corresponding bit(s) of the PFS Registers set.

Subsection 0 - Tests the MAC BZ ZIF board for reporting data errors on data received from the MCH.

Subsection 1 - Tests the MAC BZ ZIF board for reporting channel errors when illegal op codes and illegal type codes are sent to MAC.

Subsection 2 - Tests the MAC BZ and BK ZIF board parity checkers for reporting parity errors and checks that the MAC "force bad parity" circuitry functions properly (PTM bit 18).

Subsection 3 - Tests the parity error reporting on MAC Data Bus and from PFS Register Board 4.

Subsection 4 - Tests the parity error reporting from PFS Register Board 3.

Subsection 5 - Tests the parity error reporting from PFS Register Board 2.

Subsection 6 - Tests the parity error reporting from PFS Register Board 1.

Subsection 7 - Tests the parity error reporting from PFS Register Board 0.

Subsection 8 - Tests the parity error reporting from the MAC Reference ROM read data parity checker on the MAC BK ZIF Boad.

Subsection 9 - Tests the MAC BZ ZIF board reporting of channel parity errors on Control Word 1, Control Word 2 and write data. Also ensures that these type parity errors are not reported as read data parity errors.

Hardware Tested:

- Error reporting logic (parity checkers, illegal op code and illegal type code) on the BZ and BK MAC ZIF boards.
- The parity error reporting logic on all five PFS Register boards.
- The logic path (wire and circuitry) from the error detection logic to the PFS Registers' bits, for those parity checkers being verified.
- The IOU and MCH logic used for detecting channel data errors and reporting such errors in the IOU Fault Status Register 2.

Method:

Basically, each subsection sends data with bad parity down a data path and then verifies that the appropriate fault bits in the PFS Registers and/or IOU Fault Status Register have recorded the parity error.

Section 4, Subsection 0

This subsection tests the MAC BZ ZIF board for reporting data errors on data received from the MCH.

Method:

The subsection sets the IOU Test Mode Register bit 59 to force channel parity bit to 0. The subsection then sends a Master Clear Function on the MCH to MAC. The Master Clear Function code should have bad parity which should be detected by MAC's input data parity checker. The MCH channel error, IOU Fault Status Register 2, and PFS Register 86 are then read and verify that the channel data error was detected by MAC and properly reported.

Initialization:

- Clear IOU Fault Status Registers and clear MCH channel error if set.

Conditions:

C.0
a. Set IOU Environment Control Register (30 (hex) bit 59 to enable Test Mode Register and set IOU Test Mode Register (A0 (hex) bit 59 if an I2 or bit 58 if an I4, to force channel parity bit to 0 (cause bad parity on channel data).
b. Send Master Clear Function (op code equals six, Type equals zero) to MAC.
c. Report error if MCH does not go inactive before timeout of 100 microseconds by setting Condition Result to AA0000--00 (hex); expected result equals zero.

C.1 Report error if MCH does not have a channel error (no valid data to report, so report data equal to 0--0).

C.2
a. Clear IOU Environment Control Register bit 59 and IOU Test Mode Register bit 58 or 59 to cause good parity on channel data.
b. Read and Report IOU Fault Status Register 2 bytes 5, 6 and 7; error if bits 40 and 59 (bits 40 and 58 if testing CPU1) are not set. Note that this applies to I2 only. Bit 40 is clear on an I4.

C.3 Read and Report contents of PFS Register 86 (hex) bits 32 through 38 and 63, corresponding bit positions in Condition Result (report error if bit 33 is not set and bits 32, 34 through 38, and 63 are not clear).

Section 4, Subsection 1

This subsection tests the MAC BZ ZIF board for reporting channel errors when illegal op codes and illegal type codes are sent to MAC.

Method:

The subsection sends functions with illegal op codes and checks to verify that a channel error occurs. Similarly, functions with illegal type codes are sent and a check made to verify that a channel error occurs (only illegal type code equals eight is used since the hardware checks only the upper bit of the type code when sensing for illegal type codes; types 8, 9, and B through F are all illegal).

Conditions:

C.0 Master Clear processor; Op Code equals six, type code equals zero.
- a. Report error if MCH does not go inactive before timeout of 100 microseconds by setting Condition Result to AA0000--00 (hex); expected result equals zero.
- b. Report error if MCH channel error occurs.

C.1 Send Illegal Op Code Function to processor; Op Code equals two, type equals zero.
- a. Report error if MCH does not go inactive before timeout of 100 microseconds by setting Condition Result to AA0000--00 (hex); expected result equals zero.
- b. Report error if MCH channel error does not occur.

C.2 Send Illegal Op Code Function to processor; Op Code equals three, type equals zero.
- a. Report error if MCH does not go inactive before timeout of 100 microseconds by setting Condition Result to AA0000--00 (hex); expected result equals zero.
- b. Report error if MCH channel error does not occur.

C.3 Send Illegal Op Code Function to processor; Op Code equals nine, Type equals zero.
- a. Report error if MCH does not go inactive before timeout of 100 microseconds by setting Condition Result to AA0000--00 (hex); expected result equals zero.
- b. Report error if MCH channel error does not occur.

C.4 Master Clear processor; Op Code equals six, Type equals A.
- a. Report error if MCH does not go inactive before timeout of 100 microseconds by setting Condition Result to AA0000--00 (hex); expected result equals zero.
- b. Report error if MCH channel error occurs.

C.5 Send Illegal Type Code Function to processor; Op Code equals six, Type equals eight.
- a. Report error if MCH does not go inactive before timeout of 100 microseconds by setting Condition Result to AA0000--00 (hex); expected result equals zero.
- b. Report error if MCH channel error does not occur.

Section 4, Subsection 2

This subsection tests the MAC BZ and BK ZIF board parity checkers for reporting parity errors and checks that the MAC "force bad parity" circuitry functions properly.

Method:

The "force 0 parity bit in MAC" bit is set in the processor PTM Register (bit 18). Bad parity data is then sent to MAC as a Master Clear Function and as Echo data. The appropriate PFS Register bits are then checked to verify that they are set to report the parity errors from the different MAC parity checkers.

Initialization:

- Send a Master Clear Function and a Clear Error Function to processor. Clear MCH Channel Error if set.

Conditions:

C.0
- a. Set bit 18 (force 0 parity bit in MAC) of PTM Register (A0 (hex).
- b. Send Master Clear Function; Op Code equals six, Type equals one. This function should cause a function parity error in MAC.
- c. Report error if MCH Channel Error Flag set (no valid data to report, so report 0--0).

C.1
- a. Clear bit 18 of PTM Register (do write of 0--0 to PTM).
- b. Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bit 34, 35 and 63 are not set and other bits are not clear.

C.2
- a. Clear MCH Channel Error if set.
- b. Set bit 18 (force 0 parity bit in MAC) of PTM Register (A0 (hex).
- c. Send Clear Error Function to processor and IOU.
- d. Send Echo Function to processor; Op Code equals eight, Type equals seven: Activate,
  Send Control Word 1 equals zero,
  Control Word 2 equals one,
  Deactivate
  Activate,
  Input one byte,
  Deactivate.
- e. Save IOU Fault Status Register 2 contents of bytes 5, 6 and 7.
- f. Clear bit 18 of PTM Register (write 0--0's).
- g. Report data byte input; error if not equal to 01.

C.3 Report saved contents of IOU Fault Status Register 2; error if MCH channel error did not occur in C.1d, and bits 40 and 59 (bits 40 and 58 if testing CPU1) of IOU Fault Status Register 2 are not both set. Note that this applies to I2 only. Bit 40 is clear on an I4.

Conditions:

C.4 Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bits 32, 36 and 37 are not set; and all others are not clear.

C.5 Read, format and report bit 63 from PFS Registers 80, 82, 84, 86 and 88; error if all bit 63's are not clear.

Format:

Condition Result bit number equals	59	60	61	62	63
Bit 63 from PFS number	80	82	84	86	88

Section 4, Subsection 3

This subsection tests the parity error reporting on MAC Data Bus and from PFS Register Board 4.

Method:

Data with bad parity is written into byte 5 of the PTM register. This is accomplished by writing six bytes to the PTM register, all bytes containing zeros except bytes 2 and 5. Byte 2 sets bit 18 of PTM (force 0 parity bit in MAC) which then results in a 0 parity bit for byte 5 which contains 01 (hex) for data. Byte 5 of PTM now contains bad parity, and the other bytes good parity.

Bit 18 of the PTM Register is cleared by writing only three bytes of zeros to PTM, this leaves byte 5 with its bad parity.

The entire PTM Register is then read and byte 5 data should cause parity errors to be detected on PFS Register board 4 and in the MAC ZIF boards and on the MCH.

Initialization:

- Send Master Clear Function to processor.

Conditions:

C.0
a. Write PTM Register (A0 (hex) with bad parity data in byte 5 (write 6 bytes of data equal to 00, 00, 20, 00, 00, 01 (hex).
b. Clear bit 18 of PTM Register by writing three bytes of data equal to 00, 00, 00 (hex).
c. Clear Errors in processor, IOU and clear MCH channel error if set.
d. Nothing valid to report so report 0.

Conditions:

C.1 a. Read and report contents of the PTM Register; error if not 0000 0000 0001 0000 (hex).

C.2 Read and report IOU Fault Status Register 2 bytes 5, 6 and 7; report error if MCH channel error not set after reading PTM Register and bits 40 and 59 (bits 40 and 58 if testing CPU1) of IOU Fault Status Register 2 are not set. Note that this applies to I2 only. Bit 40 is clear on an I4.

C.3 Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bits 32 and 35 are not set; and other bits are not clear.

C.4 Read, format and report bit 63 from PFS Registers 80, 82, 84, 86 and 88; error if bit 63 of PFS 88 is not set and bit 63 of other PFS Registers are not clear.

Format:

Condition Result bit number equals	59	60	61	62	63
Bit 63 from PFS number	80	82	84	86	88

Section 4, Subsection 4

This subsection tests the parity error reporting from PFS Register Board 3.

Method:

Data with bad parity is written into byte 3 of the PTM register. The data is then read from the PTM register which should result in parity error reporting from PFS Register Boards 3 and 4 and from the MAC ZIF boards and the MCH.

See Section 4, Subsection 3 Method for additional detail.

Initialization:

- Send Master Clear to processor.

Conditions:

C.0
a. Write PTM Register (A0 (hex) with bad parity data in byte 3 (write 6 bytes of data equal to 00, 00, 20, 01, 00, 00 (hex).
b. Clear bit 18 of PTM Register by writing three bytes of data equal to 00, 00, 00 (hex).
c. Clear Errors in processor, IOU and clear MCH channel error if set.
d. Nothing valid to report so report 0.

Conditions:

C.1 a. Read and report contents of the PTM Register; error if not 0000 0001 0000 0000 (hex).

C.2 Read and report IOU Fault Status Register 2 bytes 5, 6 and 7; report error if MCH channel error not set after reading PTM Register and bits 40 and 59 (bits 40 and 58 if testing CPU1) of IOU Fault Status Register 2 are not set. Note that this applies to I2 only. Bit 40 is clear on an I4.

C.3 Read and Report PFS Register 86 bits 32 through 38 and 63 in the correspondng bit positions in Condition Result; error if bits 32, 35, and 63 are not set; and other bits are not clear.

C.4 Read, format and report bit 63 from PFS Registers 80, 82, 84, 86 and 88 (error if bit 63 of PFS 88 and 86 are not set and bit 63 of other PFS Registers are not clear).

Format:

Condition Result bit number equals	59	60	61	62	63
Bit 63 from PFS number	80	82	84	86	88

Section 4, Subsection 5

This subsection tests the parity error reporting from PFS Register Board 2.

Method:

Data with bad parity is written into byte 1 of the PTM register. The data is then read from the PTM register which should result in parity error reporting from PFS Register Boards 2, 3 and 4 and from the MAC ZIF boards and the MCH.

See Section 4, Subsection 3 Method for additional detail.

Initialization;

- Send Master Clear to processor.

Conditions:

C.0 a. Write PTM Register (A0 (hex) with bad parity data in byte 1 (write 6 bytes of data equal to 00, 00, 20, 00, 00, 00 (hex).
b. Clear bit 18 of PTM Register by writing three bytes of data equal to 00, 01, 00 (hex) (leaves bad parity data in byte 1).
c. Clear Errors in processor, IOU and clear MCH channel error if set.
d. Nothing valid to report so report 0.

Conditions:

C.1 a. Read and report contents of the PTM Register; error if not 0001 0000 0000 0000 (hex).

C.2 Read and report IOU Fault Status Register 2 bytes 5, 6 and 7; report error if MCH channel error not set after reading PTM Register and bits 40 and 59 (bits 40 and 58 if testing CPU1) of IOU Fault Status Register 2 are not set. Note that this applies to I2 only. Bit 40 is clear on an I4.

C.3 Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bits 32, 35, and 63 are not set; and other bits are not clear.

C.4 Read, format and report bit 63 from PFS Registers 80, 82, 84, 86 and 88; error if bit 63 of PFS 88, 86 and 84 are not set and bit 63 of other PFS Registers are not clear.

Format:

Condition Result bit number equals	59	60	61	62	63
Bit 63 from PFS number	80	82	84	86	88

- Write PTM Register with 0--0

Section 4, Subsection 6

This subsection tests the parity error reporting from PFS Register Board 1.

Method:

Data with bad parity is written into byte 7 of the DEC register. The data is then read from the DEC register which should result in parity error reporting from PFS Register Boards 1, 2, 3, and 4 and from the MAC ZIF boards and the MCH.

Initialization:

- Send Master Clear to processor.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force bad parity (zero parity bit) through MAC.
b. Write DEC Register (30) with bad parity data in byte 7 by writing 8 bytes of data equal to 00, 00, 00, 00, 00, 00, 00, 01.
c. Clear bit 18 of PTM Register by writing it with 0--0.
d. Clear Errors in processor, IOU and clear MCH channel error if set.
e. Nothing valid to report so report 0.

C.1 a. Read and report contents of the DEC Register; error if not 0000 0000 0000 0001 (hex).

C.2 Read and report IOU Fault Status Register 2 bytes 5, 6 and 7; report error if MCH channel error not set after reading DEC Register and bits 40 and 59 (bits 40 and 58 if testing CPU1) of IOU Fault Status Register 2 are not set. Note that this applies to I2 only. Bit 40 is clear on an I4.

C.3 Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bits 32, 35 and 63 are not set; and other bits are not clear.

C.4 Read, format and report bit 63 from PFS Registers 80, 82, 84, 86 and 88; error if bit 63 from PFS 82, 84, 86 and 88 are not set and bit 63 from PFS 80 is not clear.

Format:

Condition Result bit number equals	59	60	61	62	63
Bit 63 from PFS number	80	82	84	86	88

Section 4, Subsection 7

This subsection tests the parity error reporting from PFS Register Board 0.

Method:

Data with bad parity is written into byte 5 of the DEC Register. The data is then read from the DEC register which should result in parity error reporting frm PFS Register Boards 0, 1, 2, 3 and 4 and from the MAC ZIF boards and the MCH.

Initialization:

- Send Master Clear to processor.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force bad parity (zero parity bit) through MAC.
b. Write DEC Register (30) with bad parity data in byte 5 by writing 8 bytes of data equal to 00, 00, 00, 00, 00, 01, 00, 00.
c. Clear bit 18 of PTM Register by writing it with 0--0.
d. Clear Errors in processor, IOU and clear MCH channel error if set.
e. Nothing valid to report so report 0.

C.1 a. Read and report contents of the DEC Register; error if not 0000 0000 0001 0000 (hex).

C.2 Read and report IOU Fault Status Register 2 bytes 5, 6 and 7; report error if MCH channel error not set after reading DEC Register and bits 40 and 59 (bits 40 and 58 if testing CPU1) of IOU Fault Status Register 2 are not set. Note that this applies to I2 only. Bit 40 is clear on an I4.

C.3 Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bits 32, 35 and 63 are not set; and other bits are not clear.

C.4 Read, format and report bit 63 from PFS Registers 80, 82, 84, 86 and 88; error if bit 63 from PFS 80, 82, 84, 86, and 88 are not set.

Format:

Condition Result bit number equals	59	60	61	62	63
Bit 63 from PFS number	80	82	84	86	88

- Write DEC Register with 0--0.

Section 4, Subsection 8

This subsection tests the parity error reporting from the MAC Reference ROM read data parity checker on the MAC BK ZIF Board.

Method:

Data with bad parity is written into location 0 of the MAC Reference ROM. Location 0 is then read (swept) which should result in a parity error being reported from the Reference ROM read data parity checker.

Initialization:

- Send Master Clear to processor.

Conditions:

C.0 a. Set bit 18 of the PTM Register (A0) to force bad parity on data through MAC (forces zero parity bit).
b. Write Reference ROM with bad data:
Write Function; Op Code equals five, Type equals three,
Activate,
Send Control Word 1 and Control Word 2 both equal to 0--0,
Deactivate,
Activate,
Output 4 bytes of data equal to 01, 01, 01, 01,
Deactivate.
c. Clear bit 18 of the PTM Register by writing it with 0--0.
d. Clear Errors in processor.
e. No valid data is reported from this condition, so report 0--0.

C.1 a. Read (sweep) Reference ROM location 0 (ignore data).
b. Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bit 38 is not set and other bits are not clear.

Write Reference ROM location 0 with 4 bytes of 00.

Section 4, Subsection 9

This subsection tests the MAC BZ ZIF board reporting of channel parity errors on Control Word 1, Control Word 2 and write data. Also ensures that these type parity errors are not reported as read data parity errors.

Method:

The "force channel parity bit to 0" bit is set in the IOU Test Mode Register (bit 59). A MAC operation consisting of sending a write function code, Control Word 1, Control Word 2, and one write data byte is executed. Checks are then made to verify that a MCH Channel Error has been reported, the processor PFS Register 86 has recorded a channel parity error on data sent to MAC (bit 33 set), and processor PFS Register 86 has not recorded a parity error on garbage data in MAC's Channel Output Register (bit 32 clear).

The above sequence is repeated with bad parity occurring in one word of the write operation each pass function, then Control Word 1, then Control Word 2, and finally the write data byte.

Initialization:

- Send a Master Clear Function and a Clear Error Function to processor. Clear MCH Channel Error if set.

Repeat the following conditions for each test pattern sequence.

Conditions:

C.0 a. Set IOU Environment Control Register (30 (hex) bit 59 to enable the Test Mode Register.
b. Execute MAC operation sequence per test pattern sequence:
Function Code = test pattern
Send Control Word 1 = test pattern
Send Control Word 2 = test pattern
Send Data Byte = test pattern
c. Report pattern number in Condition Result byte 7; error if MCH Channel Error did not occur.

C.1 a. Clear IOU Environment Control Register bit 59 twice.
b. Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result (error if bit 32 is not clear and and bit 33 is not set, other bits are do not care).

Exit subsection if end of test patterns.

Test Pattern Sequences (Hexadecimal):

0	Function Code	= 53+	1	Function Code	= 54
	Control Word 1	= 01		Control Word 1	= 00+
	Control Word 2	= 01		Control Word 2	= 01
	Write Data	= 01		Write Data	= 01
2	Function Code	= 54	3	Function Code	= 54
	Control Word 1	= 01		Control Word 1	= 01
	Control Word 2	= 00+		Control Word 2	= 01
	Write Data	= 01		Write Data	= 00+

SECTION 5 - TEST MAC HARDWARE PARITY CHECKERS

This section verifies that the parity checkers on the MAC ZIF boards and PFS Registers boards are completely functional - do not report parity errors when they shouldn't.

Subsection 0 - Tests the parity checkers on the MAC BK and BZ ZIF boards.

Subsection 1 - Tests the parity checkers on the PFS Register boards (Boards 0, 1, 2, 3 and 4) and the MAC Mux 5 parity checker.

Hardware Tested:

- Parity checkers on the MAC BK and BZ ZIF boards excluding the Reference ROM Parity Checker which will be tested by a later section.
- The parity checker on each PFS Register Board.

+ This word of the sequence will cause bad channel parity and bad parity within MAC.

Method:

rror when data with bad parity is checked (using one data pattern), the emainder of the parity checker circuit can be tested using data patterns with ood parity. A faulty parity checker circuit will then report a parity error hen it should not.

n incrementing data pattern is used to test the parity checkers in order to uarantee thorough testing of the parity checker circuits. These 256 pattern perands are required because the exact wiring of the parity checker circuits which data bit to which pin) is unknown and varies from parity checker to arity checker.

Section 5, Subsection 0

This subsection tests the parity checkers on the MAC BK and BZ ZIF boards.

Method:

ata pattern. After each data transfer (one byte) the appropriate error tatus bits are checked to verify that a parity error did not occur. A faulty arity checker will report a parity error when it shouldn't.

Initialization:

- Send Master Clear and Clear Errors to MAC. Also, clear the MCH channel error flag if set.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Send Echo Function to MAC.
b. Activate MCH.
Send Control Word 1 equal to zero.
Send Control Word 2 equal to Test Pattern.
Deactivate MCH.
c. Report Test Pattern; informational.

C.1
a. Activate MCH.
b. Input one byte of data.
c. Deactivate MCH.
d. Report data input; error if data input does not equal test pattern.

C.2 If MCH channel error flag is set, report error and IOU Fault Status Register 2 bytes 5, 6 and 7; expected result equals zero.

C.3 Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bits not all clear.

Exit subsection if end of test patterns.

Test Patterns (Hexadecimal):

The Test Patterns consist of an incrementing 8 bit data pattern:

00
01
02
|
FD
FE
FF

Section 5, Subsection 1

This subsection tests the parity checkers on th PFS Register Boards (Boards 0, 1, 2, 3 and 4) and the MAC Mux 5 parity checker.

Method:

The subsection writes an incrementing byte pattern into the DEC Register (bytes 4, 5, 6 and 7). The contents of the DEC Register is then read. After each read of the DEC Register, the appropriate error status data is checked to verify that a parity error did not occur. A faulty parity checker will report a parity error when it should not.

Initialization

- Send Master Clear and Clear Errors to MAC. Also, clear the MCH channel error flag if set.

Repeat the following conditions for each test pattern.

Conditions:

C.0 Write the DEC Register (30) with Test Pattern.
a. Report Test Pattern written; informational.

C.1 Read the DEC Register.
a. Report data read; error if not equal to Test Pattern written.

C.2 If MCH channel error flag is set, report error and IOU Fault Status Register 2 bytes 5, 6 and 7 (expected result equals zero).

C.3 Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if any bit is set.

C.4 Read, format and report bit 63 from PFS Registers 80, 82, 84, 86 and 88; error if any bit is set.

Format:

Condition Result bit number equals	59	60	61	62	63
Bit 63 from PFS number	80	82	84	86	88

Exit subsection if end of test patterns.

Test Patterns (Hexadecimal):

The Test Patterns consist of an incrementing byte pattern in bytes 4 through 7:

byte =	0	1	2	3	4	5	6	7
1	00	00	00	00	00	01	02	03
2	00	00	00	00	04	05	06	07
3	00	00	00	00	08	09	0A	0B
63	00	00	00	00	F8	F9	FA	FB
64	00	00	00	00	FC	FD	FE	FF

SECTION 6 - TEST MAC REFERENCE ROM MEMORY

This section verifies that the MAC Reference ROM functions correctly (holds data, read data parity checker is functioning, and all addressing logic internal and external to the Reference ROM RAM chips function correctly).

Subsection 0 - Tests the MAC Reference ROM for holding data in all locations.

Subsection 1 - Tests the MAC Reference ROM read data parity checker.

Subsection 2 - Tests the Reference ROM addressing logic, both external and internal to the memory's RAM chips.

Hardware Tested:

- The Reference ROM RAM chips, input and output to/from the RAM chips, and the addressing logic internal and external to the RAM chips; all contained on the MAC BK ZIF board.

Method:

See the individual subsections Method documentation.

NOTE

Clock pulse width margins, per PARAM1, are enabled by this section for the remaining sections in this test.

Section 6, Subsection 0

This subsection tests the MAC Reference ROM for holding data in all locations.

Method:

The subsection writes the entire Reference ROM with a data pattern. The memory is then swept in its entirety, after which the parity status is checked. If any location drops or picks a bit, a parity error should occur.

Initialization:

- Send Master Clear.

Repeat the following conditions for each test pattern.

Conditions:

C.0 Write Reference ROM with Test Pattern (100 (hex) 32 bit words).
a. Report Test Pattern written. Pattern number in byte 1, data in bytes 4 through 7; informational.
b. Clear Errors in processor.

C.1 a. Sweep all of Reference ROM (100 (hex) locations).
b. Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bits are not all clear.

Exit subsection if end of test patterns.

Test Patterns (Hexadecimal):

No.	Pattern
0	00 00 00 00
1	00 00 00 3F
2	00 00 00 2A
3	00 00 00 15
4	00 00 00 00

Section 6, Subsection 1

This subsection tests the MAC Reference ROM read data parity checker.

Method:

The subsection writes an incrementing pattern throughout the Reference ROM. The memory locations are then swept. This causes the incrementing data pattern bytes to be input to the parity checker. If the parity checker is faulty, a parity error will be reported when it should not be.

Initialization:

- Send Master Clear.

Conditions:

C.0 a. Write half of Reference ROM with an incrementing data pattern (location 0 contains 00 (hex), 1 contains 01 (hex),,, 3F (hex) contains 3F (hex). No valid data, so report 0--0.
b. Clear Errors in processor.

C.1 a. Sweep half of Reference ROM.
b. Read and Report PFS Register 86 bits 32 through 38 and 63 in the corresponding bit positions in Condition Result; error if bits are not all clear.

Section 6, Subsection 2

This subsection tests the Reference ROM addressing logic, both external and internal to the memory's RAM chips.

Method:

The subsection uses a march pattern technique to test the Reference ROM addressing logic. Also, because data cannot be directly read by the MCH from the memory, the Reference ROM data parity status must be used to determine when the wrong memory location has been read. The test sequence used is outlined below:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent 2 locations (X+1 and X+2) when sweeping Reference ROM.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

Initialization:

- Send Master Clear.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0 a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1 Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2 Report Test Address X in Condition Result byte 7; informational.

C.3 a. Sweep Test Address X location in the Reference ROM.
b. Clear Errors in processor.
c. Sweep Test Address X location in the Reference ROM second time.
d. Read, format and report Parity Status; error if not 0.

If Repeat Iteration set, repeat C.3.

C.4 If C.3 did not fail,
a. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
b. Clear Errors in processor.
c. Nothing valid to report so report 0--0.

Else, (C.3 did fail) nothing valid to report so report 0--0.

epeat C.2 through C.4.

NOTE

Must skip address 86 (hex), since it is read out when reading PFS 86 (hex); and must skip address 00, since it is read on every MAC functon.

Exit subsection if error.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
01	00 00 00 5F
02	|
03	|
incrementing	Same data pattern
address	|
FD	|
FE	00 00 00 5F

Base Pattern (Hexadecimal):

0 00 00 00 20

Parity Status Format:

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	MAC Reference ROM Read Data (loc C1D)	86	38

SECTION 7 - TEST MAC ADDRESS BUS PATH

This section verifies that the MAC Address Bus path can carry addresses over its entire length without dropping or picking any bits.

Subsection 0 - Tests MAC Address Bus path with various address patterns.

Hardware Tested:

- The MAC Address Bus path from the MAC BK ZIF board to the IF Instruction Assembly Register (LSI Arrays), including the associated logic.
- The MAC Address Bus path from the MAC BK ZIF board through the ALN Soft Control Memory Auxiliary board through the two AC Soft Control Memory Auxiliary boards (SC 1 and 2) through the two BDP Port boards through the two BDP RAM Auxiliary boards to the MAC BZ ZIF board, including all associated logic.

Method:

See subsection 0 Method documentation.

Section 7, Subsection 0

Tests MAC Address Bus path with various address patterns.

Method:

The subsection reads (sweeps) a location from each of the Soft Control Memories connected to the MAC Address Bus path. When sweeping a Soft Control Memory location the address is returned to the MCH as data, after passing the entire length of the MAC Address Bus path. Any bit drops or picks will be evident in the data (address) read by the MCH.

Note that MAC will send the subsequent address out even though only 1 location is desired, in effect sweeping 2 locations. Therefore, if a parity error occurs, C.3 will sweep the subsequent location and report the data read by MCH.

In order to isolate a failure in the MAC Address Bus to three or less replaceable subassemblies, the parity status from each Soft Control Memory's Address Register is reported. By analysis of this parity status, the point of failure within the bus can be determined.

Note that this subsection is not checking the contents of the memories, only the address sent to each of the memories.

This subsection uses various address data patterns to ensure complete testing of the MAC Address Bus path.

Initialization:

- Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each address pattern X.

Conditions:

C.0
a. Sweep location X and X+1 in IF C180 Decode Memory, ignore input data.
b. Sweep location X and X+1 in BDP Port A RAM, ignore input data.
c. Sweep location X and X+1 in BDP Port B RAM, ignore input data.
d. Sweep location X and X+1 in BDP RAM PAK 0, EBCDIC Memory, ignore input data.
e. Sweep location X and X+1 in BDP RAM PAK 0, Convert Memory, ignore input data.
f. Sweep location X and X+1 in BDP RAM PAK 1, Spec Error Memory, ignore input data.
g. Report address pattern X, pattern number in byte 1 and pattern data in byte 7 of Condition Result; informational.

C.1
a. Sweep location X in BDP RAM PAK 1, X256 memory, and input address from MAC Address Bus.
b. Report data input (address) when location X swept; error if data is not equal to address pattern X.

C.2 Read, format and report appropriate parity status bits; error if bits are not all clear.

If C.2 failed, execute C.3; else skip C.3.

C.3
a. Sweep location X+1 in BDP RAM PAK 1, and input address from MAC Address Bus.
b. Report data input (address) when location X+1 swept, error if data is not equal to address pattern X+1.

Exit subsection if end of address patterns.

Address Patterns X (Hexadecimal):

0 00 (parity bit equals zero)

1 FF (parity bit equals zero)

2 55 (parity bit equals zero)

3 AA (parity bit equals zero)

4 DF (parity bit equals one)

5 FD (parity bit equals one)

Parity Status Format:

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	MAC Channel Out Register (loc C1E)	86	32
62	BDP RAM PAK 1 (loc A3B2)	81	2
61	BDP RAM PAK 0 (loc A3A2)	81	10
60	BDP RAM PAK 0 (loc A3A2)	81	8
59	BDP Port B Board (loc A3C1)	80	54
58	BDP Port A Board (loc A3B1)	80	55
57	IF Decode Address Reg. (LSI Arrays)	88	1
56	IF Decode Address Reg. (LSI Arrays)	88	0
55	MAC Address (loc C1D)	86	37

SECTION 8 - TEST AC/ALN SCM PAKS ERROR REPORTING AND BYTE SELECTS

This section tests each AC and ALN Soft Control Memory Auxiliary board to ensure that their parity checkers report parity errors properly. In addition, the MAC Bus Byte Select paths to and through these SCM boards is also verified as operational.

Subsection 0 - Tests parity error reporting from AC Soft Control 2 and MAC Bus Byte Select path to AC Soft Control 2.

Subsection 1 - Tests parity error reporting from AC Soft Control 1 and MAC Bus Byte Select path through AC Soft Control 2 to AC Soft Control 1.

Subsection 2 - Tests parity error reporting from ALN Soft Control PAK and MAC Bus Byte Select path through AC Soft Control 1 to ALN Soft Control PAK.

Hardware Tested:

- The MAC Bus Byte Select logic on the MAC BK ZIF board, including a partial check of the MAC byte counter.
- The MAC Bus Byte Select path from the BK MAC ZIF board through the AC Soft Control 1 and 2 PAKs to the ALN Soft Control PAK.
- The parity error reporting for memory data parity errors from each of the AC Soft Control PAKS and the ALN Soft Control PAK.
- The corresponding parity error status bits in the PFS Registers PFS Register 80 bits 46 and 45; PFS Register 88 bit 56).
- The byte select logic on the AC and ALN Soft Control PAKs.

Method:

The same testing sequence is used in all of this section's subsections, only the Soft Control (SC) select bits are different.

Data is written into location 0 of the SC. Only one byte of this 4 byte data contains bad parity. Location 0 is then read (sweep) and the PFS Register parity status is checked to verify that a parity error has been reported.

Four data patterns are used, each with a different byte containing bad parity. If the byte select logic is faulty, one of the data patterns will not generate a parity error because a good parity data byte will overwrite the bad parity byte.

Test Patterns:

Same for all subsections:

0	01	00	00	00 (hex)	Bad parity in byte 0
1	00	01	00	00 (hex)	Bad parity in byte 1
2	00	00	01	00 (hex)	Bad parity in byte 2
3	00	00	00	01 (hex)	Bad parity in byte 3

Parity Status Format:

Same for all subsections:

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	AC Soft Control 1 (loc A2G1)	80	46
62	AC Soft Control 2 (loc A2E1)	80	45
61	ALN Soft Control PAK (loc C2K1)	80	44

Section 8, Subsection 0

This subsection tests parity error reporting from AC Soft Control 2 and MAC Bus Byte Select path to AC Soft Control 2.

Method:

See Section 8 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of AC Soft Control 2 with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1 Read (sweep) AC Soft Control 2 location 0 and report data read; error if not equal to 40 (hex).

C.2 Read, format and report AC/ALN parity status bits from PFS Registers 80 and 88; error if bit 45 of PFS 80 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 through C.2 if more test patterns.

C.3 a. Write location 0 of AC Soft Control 2 with 0--0.
b. Sweep location 0 of AC Soft Control 2.
c. Clear Errors in processor.
d. Sweep location 0 of AC Soft Control 2.
e. Read, format and report AC/ALN parity status bits from PFS Registers 80 and 88; error if all bits are not clear.

Test Patterns:
See Section 8 Description.

Parity Status Format:
See Section 8 Description.

Section 8, Subsection 1

This subsection tests parity error reporting from AC Soft Control 1 and MAC Bus Byte Select path through AC Soft Control 2 to AC Soft Control 1.

Method:
See Section 8 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of AC Soft Control 1 with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1 Read (sweep) AC Soft Control 1 and report data read; error if not equal to 80 (hex).

C.2 Read, format and report AC/ALN parity status bits from PFS Registers 80 and 88; error if bit 46 of PFS 80 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 through C.2 if more test patterns.

C.3 a. Write location 0 of AC Soft Control 1 with 0--0.
b. Sweep location 0 of AC Soft Control 1.
c. Clear Errors in processor.
d. Sweep location 0 of AC Soft Control 1.
e. Read, format and report AC/ALN parity status bits from PFS Registers 80 and 88; error if all bits are not clear.

Test Patterns:

See Section 8 Description.

Parity Status Format:

See Section 8 Description.

Section 8, Subsection 2

This subsection tests parity error reporting from ALN Soft Control PAK and MAC Bus Byte Select path through AC Soft Control 1 to ALN Soft Control PAK.

Method:

See Section 8 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of ALN Soft Control PAK with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1 Read (sweep) ALN Soft Control PAK and report data read; error if not equal to C0 (hex).

C.2 Read, format and report AC/ALN parity status bits from PFS Registers 80 and 88; error if bit 56 of PFS 88 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 through C.2 if more test patterns.

C.3 a. Write location 0 of ALN Soft Control PAK with 0--0.
b. Sweep location 0 of ALN Soft Control PAK.
c. Clear Errors in processor.
d. Sweep location 0 of ALN Soft Control PAK.
e. Read, format and report AC/ALN parity status bits from PFS Registers 80 and 88; error if all bits are not clear.

Test Patterns:

See Section 8 Description.

Parity Status Format:

See Section 8 Description.

SECTION 9 - TEST BDP RAM PAKS ERROR REPORTING AND BYTE SELECTS

This section tests the BDP Port A and Port B Auxiliary boards and the BDP RAM PAK Auxiliary boards to ensure that their RAM parity checkers report parity errors properly. In addition, the MAC Bus Byte Select paths to and through these auxiliary boards are also verified as operational.

Subsection 0 - Test RAM parity error reporting from the BDP Port A board and MAC Bus Byte Select path through AC Soft Control 2 PAK to the BDP Port A board.

Subsection 1 - Test RAM parity error reporting from the BDP Port B board and MAC Bus Byte Select path through the BDP Port A board to the BDP Port B board.

Subsection 2 - Test RAM parity error reporting from the BDP RAM PAK 0, EBCDIC RAM, and the MAC Bus Byte Select path through the BDP Port B board to the BDP RAM PAK 0 board.

Subsection 3 - Test RAM parity error reporting from the BDP RAM PAK 0, Convert RAM.

Subsection 4 - Test RAM parity error reporting from the BDP RAM PAK 1, Spec Error RAM, and the MAC Bus Byte Select path through the BDP RAM PAK 0 board to the BDP RAM PAK 1 board.

Subsection 5 - Test RAM parity error reporting from the BDP RAM PAK 1, X256 RAM.

Hardware Tested:

- The MAC Bus Byte Select path from the AC Soft Control 2 Auxiliary board through the BDP Port A, Port B board and the BDP RAM PAK 0 board to the BDP RAM PAK 1 board.
- The individual MAC BDP Go signals originating on the MAC BZ ZIF board and sent to each BDP board.
- The parity error reporting for memory data parity errors from each of the BDP memory boards (Port A, Port B, RAM PAK 0, and RAM PAK 1).
- The corresponding parity status bits in the PFS Registers (PFS Register 81 bits 3, 4, 5, 9 and 11).
- The byte select logic on the BDP Port A and Port B boards, and the BDP RAM PAK boards.

Method:

The same testing sequence is used in all of this section's subsections, only the BDP board select bits are different.

Data is written into location 0 of the memory RAMs. Only one byte of this 4 byte data contains bad parity. Location 0 is then read (Sweep) and the PFS Register parity status is checked to verify that a parity error has been reported.

Three data patterns are used for subsections 0, 1, 3 and 5; each with a different byte containing bad parity. Only two data patterns are used for subsections 2 and 4, each with a different byte containing bad parity. If the byte select logic is faulty, one of the data patterns will not generate a parity error because a good parity data byte will overwrite the bad parity byte.

Test Patterns:

Used for all subsections, as indicated:

0	00 00 00 01 (hex) bad parity byte 3	sub 0,1,2,3,4,5
1	00 00 02 00 (hex) bad parity byte 2	sub 0,1,2,3,4,5
2	00 01 00 00 (hex) bad parity byte 1	sub 0,1,3,5

Parity Status Format:

Same for all subsections:

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	BDP RAM PAK 0 (loc A3A2)	81	11
62	BDP RAM PAK 0 (loc A3A2)	81	9
61	BDP Port B Board (loc A3C1)	81	5
60	BDP Port A Board (loc A3B1)	81	4
59	BDP RAM PAK 1 (loc A3B2)	81	3

Section 9, Subsection 0

This subsection tests RAM parity error reporting from the BDP Port A board and MAC Bus Byte Select path through AC Soft Control 1 PAK to the BDP Port A board.

Method:

See Section 9 description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of BDP Port A board RAM with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1
a. Read (Sweep) BDP Port A board RAM location 0.
b. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if bit 4 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 and C.1 if more test patterns.

C.2
a. Write location 0 of BDP Port A board RAM with 0--0.
b. Sweep location 0 of BDP Port A board RAM.
c. Clear Errors in processor.
d. Sweep location 0 of BDP Port A board RAM.
e. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if all bits are not clear.

Test Patterns:

See Section 9 Description.

Parity Status Format:

See Section 9 Description.

Section 9, Subsection 1

This subsection tests RAM parity error reporting from the BDP Port B board and MAC Bus Byte Select path through the BDP Port A board to the BDP Port B board.

Method:

See Section 9 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of BDP Port B board RAM with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1
a. Read (Sweep) BDP Port B board RAM location 0.
b. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if bit 5 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 and C.1 if more test patterns.

C.2
a. Write location 0 of BDP Port B board RAM with 0--0.
b. Sweep location 0 of BDP Port B board RAM.
c. Clear Errors in processor.
d. Sweep location 0 of BDP Port B board RAM.
e. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if all bits are not clear.

Test Patterns:

See Section 9 Description.

Parity Status Format:

See Section 9 Description.

Section 9, Subsection 2

This subsection tests RAM parity error reporting from the BDP RAM PAK 0, EBCDIC RAM, and the MAC Bus Byte Select path through the BDP Port B board to the BDP RAM PAK 0 board.

Method:

See Section 9 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of BDP RAM PAK 0, EBCDIC RAM with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1
a. Read (Sweep) BDP RAM PAK 0, EBCDIC RAM location 0.
b. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if bit 9 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 and C.1 if more test patterns.

C.2
a. Write location 0 of BDP RAM PAK 0, EBCDIC RAM with 0--0.
b. Sweep location 0 of BDP RAM PAK 0, EBCDIC RAM.
c. Clear Errors in processor.
d. Sweep location 0 of BDP RAM PAK 0, EBCDIC RAM.
e. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if all bits are not clear.

Test Patterns:

See Section 9 Description.

Parity Status Format:

See Section 9 Description.

Section 9, Subsection 3

This subsection tests RAM parity error reporting from the BDP RAM PAK 0, Convert RAM.

Method:

See Section 9 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of BDP RAM PAK 0, Convert RAM with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1
a. Read (Sweep) BDP RAM PAK 0, Convert RAM location 0.
b. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if bit 11 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 and C.1 if more test patterns.

C.2
a. Write location 0 of BDP RAM PAK 0, Convert RAM with 0--0.
b. Sweep location 0 of BDP RAM PAK 0, Convert RAM.
c. Clear Errors in processor.
d. Sweep location 0 of BDP RAM PAK 0, Convert RAM.
e. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if all bits are not clear.

Test Patterns:

See Section 9 Description.

Parity Status Format:

See Section 9 Description.

Section 9, Subsection 4

This subsection tests RAM parity error reporting from the BDP RAM PAK 1, Spec Error RAM, and the MAC Bus Byte Select path through the BDP RAM PAK 0 board to the BDP RAM PAK 1 board.

Method:

See Section 9 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of BDP RAM PAK 1, Spec Error RAM with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1
a. Read (Sweep) BDP RAM PAK 1, Spec Error RAM location 0.
b. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if bit 3 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 and C.1 if more test patterns.

C.2
a. Write location 0 of BDP RAM PAK 1, Spec Error RAM with 0--0.
b. Sweep location 0 of BDP RAM PAK 1, Spec Error RAM.
c. Clear Errors in processor.
d. Sweep location 0 of BDP RAM PAK 1, Spec Error RAM.
e. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if all bits are not clear.

Test Patterns:

See Section 9 Description.

Parity Status Format:

See Section 9 Description.

Section 9, Subsection 5

This subsection tests RAM parity error reporting from the BDP RAM PAK 1, X256 RAM.

Method:

See Section 9 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of BDP RAM PAK 1, X256 RAM with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1
a. Read (Sweep) BDP RAM PAK 1, X256 RAM location 0.
b. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if bit 3 is not set and other bits are not clear.

If Repeat Iteration set, go to C.0.

Repeat C.0 and C.1 if more test patterns.

C.2
a. Write location 0 of BDP RAM PAK 1, X256 RAM with 0--0.
b. Sweep location 0 of BDP RAM PAK 1, X256 RAM.
c. Clear Errors in processor.
d. Sweep location 0 of BDP RAM PAK 1, X256 RAM.
e. Read, format and report BDP Port and RAM PAK parity status bits from PFS Register 81; error if all bits are not clear.

Test Patterns:

See Section 9 Description.

Parity Status Format:

See Section 9 Description.

SECTION 10 - TEST IF DECODE RAMS ERROR REPORTING AND BYTE SELECTS

This section tests the IF Decode RAM auxiliary boards to ensure that their RAM parity checkers report parity errors properly. In addition, the MAC Channel Byte Count bits (1 and 2) paths to the IF Decode RAM boards are verified as operational.

Subsection 0 - Test RAM parity error reporting from the IF C170 Decode RAMs.

Subsection 1 - Test RAM parity error reporting from the IF C180 Decode RAMs.

Hardware Tested:

- The MAC Channel Byte Count bits 1 and 2 paths from the MAC BZ ZIF board to the IF Decode RAMs.
- The individual MAC Go Write signals originating on the MAC BZ ZIF board and sent to IF Control.
- The parity error reporting for memory data parity errors from the IF Decode RAMs.
- The corresponding parity status bits in the PFS Registers (PFS Register 86 bits 40, 41, 42, 48, 49 and 50).
- The byte select logic in the IF hardware.

Method:

The same testing sequence is used in both of this section's subsections, only the IF Decode RAM select bits are different (C170 Decode RAMs versus C180 Decode RAMs).

Data is written into location 0 of the Decode RAMs. Only one byte of this 4 byte data contains bad parity. Location 0 is then read (sweep) and the PFS Register parity status is checked to verify that a parity error has been reported.

Four data patterns are used for subsection 0 and two data patterns are used for subsection 1, each pattern contains a different byte with bad parity. If the byte select logic is faulty, one of the data patterns will not generate a parity error because a good parity data byte will overwrite a bad parity byte.

Note that each subsection tests a different type IF Decode RAM (C170 and C180 memories). Each memory type is spread over the same two auxiliary boards. C170 data bytes 0 and 2 on PAK 0 and bytes 1 and 3 on PAK 1. C180 data byte 0 on PAK 0 and byte 1 on PAK 1.

Parity Status Format:

Same for both subsections:

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	IF Decode PAK 0 (loc C2A1)	86	42
62	IF Decode PAK 0 (loc C2A1)	86	41
61	IF Decode PAK 0 (loc C2A1)	86	40
60	IF Decode PAK 1 (loc C2B1)	86	50
59	IF Decode PAK 1 (loc C2B1)	86	49
58	IF Decode PAK 1 (loc C2B1)	86	48

Section 10, Subsection 0

This subsection tests RAM parity error reporting from the IF C170 Decode RAMs.

Method:

See Section 10 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of IF C170 Decode RAMs with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1
a. Read (Sweep) IF CY 170 Decode RAMs location 0.
b. Report expected parity status for pattern being used; informational.

C.2 Read, format and report IF Decode RAMs parity status bits from PFS Register 86; error if parity status read is not equal expected status.

Conditions:

If Repeat Iteration set, go to C.0.

Repeat C.0 through C.2 if more test patterns.

C.3
a. Write location 0 of IF C170 Decode RAMs with 0--0.
b. Sweep location 0 of IF C170 Decode RAMS.
c. Clear Errors in processor.
d. Sweep location 0 of IF C170 Decode RAMs.
e. Read, format and report IF Decode RAMs parity status bits from PFS Register 86; error if all bits are not clear.

Test Patterns:

	Pattern	Expected Status (Formatted)
0	01 00 00 00 (hex) bad parity byte 0	0002 (hex) (PFS Reg 86, bit 41 set)
1	00 01 00 00 (hex) bad parity byte 1	0010 (hex) (PFS Reg 86, bit 49 set)
2	00 00 01 00 (hex) bad parity byte 2	0004 (hex) (PFS Reg 86, bit 40 set)
3	00 00 00 01 (hex) bad parity byte 3	0020 (hex) (PFS Reg 86, bit 48 set)

Parity Status Format:

See Section 10 Description.

Section 10, Subsection 1

This subsection tests RAM parity error reporting from the IF C180 Decode RAMs.

Method:

See Section 10 Description Method.

Initialization:

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write location 0 of IF C180 Decode RAMs with Test Pattern.
c. Clear bit 18 of PTM Register.
d. Clear Errors in processor.
e. Report Test Pattern number in byte 1 and Test Pattern data in bytes 4 through 7 of Condition Result; informational.

C.1
a. Read (Sweep) IF C180 Decode RAMs location 0.
b. Report expected parity status for pattern being used; informational.

C.2 Read, format and report IF Decode RAMs parity status bits from PFS Register 86; error if parity status read is not equal expected status.

If Repeat Iteration set, go to C.0.

Repeat C.0 through C.2 if more test patterns.

C.3
a. Write location 0 of IF C180 Decode RAMs with 0--0.
b. Sweep location 0 of IF C180 Decode RAMS.
c. Clear Errors in processor.
d. Sweep location 0 of IF C180 Decode RAMs.
e. Read, format and report IF Decode RAMs parity status bits from PFS Register 86; error if all bits are not clear.

Test Patterns:

	Pattern	Expected Status (Formatted)
0	01 00 00 00 (hex) bad parity byte 0	0001 (hex) (PFS Reg 86, bit 42 set)
1	00 01 00 00 (hex) bad parity byte 1	0008 (hex) (PFS Reg 86, bit 50 set)

Parity Status Format:

See Section 10 Description.

SECTION 11 - TEST MAC ADDRESS BUS ERROR REPORTING

This section verifies that the parity checkers along the MAC Address Bus data path will report parity errors properly.

Subsection 0 - Test parity error reporting from the IF Instruction Assembly Register (bits 0 through 7 and 8 through 15) parity checkers, from the BDP Port A and Port B boards' Stage 1 Register parity checkers, and from the BDP RAM PAK 0 and PAK 1 Address Register parity checkers.

Subsection 1 - Test the parity error reporting from the BDP Port A and Port B boards' Stage 2 Register (bits 0 through 7) parity checkers.

Subsection 2 - Test the parity error reporting from the BDP RAM PAK 0 and PAK 1 RAM Data Register (bits 0 through 7) parity checkers, EBCDIC and Spec Error RAMs.

Subsection 3 - Test the parity error reporting from the BDP RAM PAK 0 and PAK 1 RAM Data Registers (bits 0 through 7) parity checkers, Convert and X256 RAMs.

Hardware Tested:

- The parity error reporting from the IF Instruction Assembly Register (bits 0 through 7 and 64 and 8 through 15 and 65) parity checkers.
- The parity error reporting from the BDP Port A and Port B boards' Stage 1 Register parity checkers.
- The parity error reporting from the BDP Port A and Port B boards' Stage 2 Register (bits 27 through 34 and 35) parity checkers.
- The parity error reporting from the BDP RAM PAK 0 and PAK 1 Address Register parity checkers.
- The parity error reporting from the BDP RAM PAK 0 and PAK 1 RAM Data Register (bits 0 through 7 and 32) parity checkers.
- The corresponding parity status bits in the PFS Registers (PFS Register 80 bit 55; PFS Register 81 bits 2, 3, 4, 5, 8, 9, and 10; PFS Register 88 bits 0 and 1).

Method:

See the individual subsections Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	IF Inst Assy bits 8 through 15/65 (LSI 0B61)	88	1
62	IF Inst Assy bits 0 through 7/64 (LSI 0B60)	88	0
61	BDP Port A RAM Address (loc A3B1)	80	55
60	BDP Port B RAM Address (loc A3C1)	80	54
59	BDP Port B Stage 2 Rgtr (loc A3C1)	81	5
58	BDP Port A Stage 2 Rgtr (loc A3B1)	81	4
57	BDP RAM PAK 0-B Data (loc A3A2)	81	11
56	BDP RAM PAK 0-B Address (loc A3A2)	81	10

Condition Result Bit	Parity Error At	PFS Register No.	Bit
55	BDP RAM PAK 0-A Data (loc A3A2)	81	9
54	BDP RAM PAK 0-A Address (loc A3A2)	81	8
53	BDP RAM PAK 1 Data (loc A3B2)	81	3
52	BDP RAM PAK 1 Address (loc A3B2)	81	2

Section 11, Subsection 0

This subsection tests parity error reporting from the IF Instruction Assembly Register (bits 0 through 7 and 8 through 15) parity checkers.

Method:

One location from the IF C180 Decode RAM is read (sweep) using address data with bad parity. This bad address data is sent down the MAC Address Bus path where it is parity checked by the IF hardware. The parity status bits from the PFS Register are then read and verified that they have been set to indicate address parity errors in the IF hardware only.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on address data through MAC.
b. Sweep location 1 of IF C180 Decode RAM; Control Word 1 equals zero, Control Word 2 equals one.
c. Report sweep address read from MAC Address Bus; error if not equal 01 (hex).

Conditions:

C.1 a. Clear bit 18 of PTM Register.
b. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks.
c. Report error if the formatted status is not as follows: PFS register 88 bit 0 set, all other bits clear; Condition Result Expected equals 0--0 0002 (hex).

C.2 a. Sweep location 1 of IF C180 Decode RAM to clear out bad MAC Address Bus data.
b. Clear Errors in processor.
c. Set bit 18 of PTM Register (A0) to force 0 parity bit on address data through MAC.
d. Sweep location 1 of IF C170 Decode RAM; Control Word 1 equals one, Control Word 2 equals one.
e. Report Sweep address read from MAC Address Bus; error if not equal 01 (hex).

C.3 a. Clear bit 18 of PTM Register.
b. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks.
c. Report error if formatted status is not as follows: PFS register 88 bit 0 and 1 set, all other bits clear; Condition Result Expected equals 0--0 0003 (hex).

C.4 a. Sweep location 1 of IF C170 Decode RAM to clear out bad MAC Address Bus data.
b. Clear Errors in processor.
c. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks; error if all status bits are not clear.

Parity Status Format:

See Section 11 Description.

Section 11, Subsection 1

This subsection tests the parity error reporting from the BDP Port A and Port B boards' Stage 2 Register (bits 27 through 34) and Data Register (bits 0 through 7) parity checkers.

Method:

One location from the BDP Port A RAM and one location from the BDP Port B RAM is read (Sweep) using address data with bad parity. This bad address data is gated into the Port boards' Stage 2 Registers (bits 0 through 7). The parity status bits from the PFS Registers are then read and verified that they have been set to indicate parity errors from the Stage 2 Register parity checkers for bits 0 through 7, and the Stage 1 Register parity checkers for bits 0 through 7.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on address data through MAC.
b. Sweep location 1 of BDP Port A RAM; Control Word 1 equals zero, Control Word 2 equals one.
c. Report sweep address read from MAC Address Bus; error if not equal 01 (hex).

C.1 a. Clear bit 18 of PTM Register.
b. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks.
c. Report error if the formatted status is not as follows: PFS register 80 bit 55 and PFS 81 bit 4 set, all other bits clear. [Condition Result Expected equals 0--0 0024 (hex).]

C.2 a. Sweep location 1 of BDP Port A RAM to clear out bad MAC Address Bus data.
b. Clear Errors in processor.
c. Set bit 18 of PTM Register (A0) to force 0 parity bit on address data through MAC.
d. Sweep location 1 of BDP Port B RAM; Control Word 1 equals one, Control Word 2 equals one.
e. Report Sweep address read from MAC Address Bus; error if not equal 01 (hex).

C.3 a. Clear bit 18 of PTM Register.
b. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks.
c. Report error if formatted status is not as follows: PFS register 80 bit 54 and PFS 81 bit 5 set, all other bits clear. (Condition Result Expected equals 0--0 0018 (hex).)

C.4 a. Sweep location 1 of BDP Port B RAM to clear out bad MAC Address Bus data.
b. Clear Errors in processor.
c. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks; error if all status bits are not clear.

Parity Status Format:

See Section 11 Description.

Section 11, Subsection 2

This subsection tests the parity error reporting from the BDP RAM PAK 0 and PAK 1 RAM Data Registers (bits 0 through 7) parity checkers, EBCDIC and Spec Error RAMs.

Method:

One location from the BDP RAM PAK 0 (EBCDIC RAM) and one location from the BDP RAM PAK 1 (Spec Error RAM) is read (Sweep) using address data with bad parity. This bad address data is gated into the RAM PAKs' RAM Data Registers (bits 0 through 7). The parity status bits from the PFS Registers are then read and verified that they have been set to indicate parity errors from the Address Register and RAM Data Register parity checkers for bits 0 through 7.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on address data through MAC.
b. Sweep location 1 of BDP RAM PAK 0, EBCDIC RAM; Control Word 1 equals two, Control Word 2 equals one.
c. Report sweep address read from MAC Address Bus; error if not equal 01 (hex).

C.1 a. Clear bit 18 of PTM Register.
b. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks.
c. Report error if the formatted status is not as follows: PFS register 81 bits 8 and 9 set, all other bits clear. [Condition Result Expected equals 0--0 0300 (hex).]

C.2 a. Sweep location 1 of BDP RAM PAK 0, EBCDIC RAM, to clear out bad MAC Address Bus data.
b. Clear Errors in processor.
c. Set bit 18 of PTM Register (A0) to force 0 parity bit on address data through MAC.
d. Sweep location 1 of BDP RAM PAK 1, Spec Error RAM; Control Word 1 equals four, Control Word 2 equals one.
e. Report Sweep address read from MAC Address Bus; error if not equal 01 (hex).

C.3 a. Clear bit 18 of PTM Register.
b. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks.
c. Report error if formatted status is not as follows: PFS register 81 bits 2 and 3 set, all other bits clear. (Condition Result Expected equals 0--0 0C00 (hex).

Conditions:

C.4 a. Sweep location 1 of BDP RAM PAK 1, Spec Error RAM, to clear out bad MAC Address Bus data.
b. Clear Errors in processor.
c. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks; error if all status bits are not clear.

Parity Status Format:

See Section 11 Description.

Section 11, Subsection 3

This subsection tests the parity error reporting from the BDP RAM PAK 0 and PAK 1 RAM Data Register (bits 0 through 7) parity checkers, Convert and X256 RAMs.

Method:

One location from the BDP RAM PAK 0 (Convert RAM) and one location from the BDP RAM PAK 1 (X256 RAM) is read (Sweep) using address data with bad parity. This bad address data is gated into the RAM PAKs' RAM Data Registers (bits 0 through 7). The parity status bits from the PFS Registers are then read and verified that they have been set to indicate parity errors from the Address Register and RAM Data Register parity checkers for bits 0 through 7.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on address data through MAC.
b. Sweep location 1 of BDP RAM PAK 0, Convert RAM (Control Word 1 equals three, Control Word 2 equals one).
c. Report sweep address read from MAC Address Bus; error if not equal 01 (hex).

C.1 a. Clear bit 18 of PTM Register.
b. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks.
c. Report error if the formatted status is not as follows: PFS register 81 bit 10 set, all other bits clear. (Condition Result Expected equals 0--0 0080 (hex).

Conditions:

C.2 a. Sweep location 1 of BDP RAM PAK 0, Convert RAM, to clear out bad MAC Address Bus data.
b. Clear Errors in processor.
c. Set bit 18 of PTM Register (A0) to force 0 parity bit on address data through MAC.
d. Sweep location 1 of BDP RAM PAK 1, X256 RAM (Control Word 1 equals five, Control Word 2 equals one).
e. Report Sweep address read from MAC Address Bus; error if not equal 01 (hex).

C.3 a. Clear bit 18 of PTM Register.
b. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks.
c. Report error if formatted status is not as follows: PFS register 81 bit 2 set, all other bits clear. (Condition Result Expected equals 0--0 0800 16)

C.4 a. Sweep location 1 of BDP RAM PAK 1, X256 RAM, to clear out bad MAC Address Bus data.
b. Clear Errors in processor.
c. Read, format and report IF and BDP parity status for MAC Address Bus data parity checks; error if all status bits are not clear.

Parity Status Format:

See Section 11 Description.

SECTION 12 - TEST IF C180 AND C170 DECODE RAM MEMORIES

This section completely verifies the data holding and addressing capabilities of the IF Decode RAM memory logic. In addition, a verification of the MAC Address Incrementer and Parity Predictor logic is accomplished.

Subsection 0 - Test MAC Address Incrementer and Parity Predictor, and IF Instruction Assembly Register Parity Checkers for bits 0 through 7 and 64 and 8 through 15 and 65.

Subsection 1 - Test IF C180 Decode RAM memory chips with various data patterns, all locations.

Subsection 2 - Test IF C170 Decode RAM memory chips with various data patterns, all locations.

Subsection 3 - Test the IF C180 Decode RAM read data parity checkers.

Subsection 4 - Test the IF C170 Decode RAM read data parity checkers.

Subsection 5 - Test the IF C180 Decode RAM addressing logic.

Subsection 6 - Test the IF C170 Decode RAM addressing logic.

Subsection 7 - Test the addressing circuitry internal to the IF C180 Decode RAM memory chips.

Subsection 8 - Test the addressing circuitry internal to the IF C170 Decode RAM memory chips.

Hardware Tested:

- The MAC Address Incrementer and Address Incrementer Parity Predictor on the MAC BK ZIF board.
- The IF DECODE RAM's Instruction Assembly Register Parity Checkers for bits 0 through 7 and 64 and 8 through 15 and 65 which are on LSI Arrays.
- The IF DECODE RAM memory chips contained on the IF Decode RAM boards; 7CEH auxiliary boards at C2A1 and C2B1.
- The IF Decode RAM memory chips read data parity checkers on the IF Decode RAM Auxiliary boards.
- The IF Decode RAM addressing logic; including bits 0 through 7 of the Parcel Select Multiplexers (LSI Arrays), the address fanout logic on the IF Decode RAM Auxiliary boards and the address circuitry internal to each IF Decode RAM memory chip (1X256 bit chips).

Method:

See the individual subsections' Method: documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	IF C180 RAM Data (loc C2B1)	86	50
62	IF C170E RAM Data (loc C2B1)	86	49
61	IF C170O RAM Data (loc C2B1)	86	48
60	IF C180 RAM Data (loc C2A1)	86	42
59	IF C170E RAM Data (loc C2A1)	86	41
58	IF C170O RAM Data (loc C2A1)	86	40
57	IF Inst Assy Rgtr bits 8 through 15 (LSI 0B61)	88	1
56	IF Inst Assy Rgtr bits 0 through 7 (LSI 0B60)	88	0

Section 12, Subsection 0

This subsection tests MAC Address Incrementer and Parity Predictor, and IF Instruction Assembly Register Parity Checkers for bits 0 through 7 and 64 and 8 through 15 and 65.

Method:

A sweep operation is performed on the IF C180 Decode RAM and the IF C170 Decode RAM, locations 0 through FF (hex). As each location is swept, the address is read from the MAC Address Bus by the MCH and verified as being the correct address. The addresses should increment from 00 to FF (hex). If a bad address is detected, the MAC Address Incrementer should be suspected.

In addition, after the sweep of the IF Decode RAMs, the parity status for the Instruction Assembly Register, bits 0 through 7 and 64 and 8 through 15 and 65, is read from the PFS Register (88) and verified that the corresponding parity checkers did not report a parity error. A parity error reported from either parity checker could be caused by bad parity prediction from the MAC Address Incrementer Parity Predictor or a faulty IF Instruction Assembly parity checker.

The MAC Address Bus path has previously been verified as operational, so its logic components should not be suspect at this time.

Initialization:

- Send Master Clear and Clear Errors to processor. Also write all of IF Decode RAM locations (C180 and C170) with 0--0.

Conditions:

C.0
a. Send Sweep IF C180 Decode RAM function to processor, start at location 0.
b. Initialize expected address to 0.
c. Input address for one location in IF C180 Decode RAM.
d. Report address input; error if address is not equal to expected address.

If input address is equal to expected address, increment expected address and repeat steps c and d until expected address is FF (hex) (test addresses 00 through FF (hex).

C.1 Report expected address; informational.

C.2 Read, format and report IF Parity Status; error if not 0--0

C.3
a. Clear Errors in processor.
b. Send Sweep IF C170 Decode RAM function to processor, start at location 0.

c. Initialize expected address to 0.
d. Input address for one location; IF C170 Decode RAM.
e. Report address input; error if address is not equal expected address.

If input address is equal to expected address, increment expected address and repeat steps d and e until expected address is FF (hex) (test addresses 00 through FF (hex).

C.4 Report expected address; informational.

C.5 Read, format and report IF Parity Status; error if not 0--0.

Parity Status Format:

See Section 12 Description.

Section 12, Subsection 1

This subsection tests IF C180 Decode RAM memory chips with various data patterns, all locations.

Method:

All locations in the IF C180 Decode RAM memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Register 86 and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the IF C180 Decode RAM locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.
c. Clear Errors in processor.

C.1
a. Sweep all locations of IF C180 Decode RAM (locations 00 through FF (hex).
b. Read, format and report IF Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 7F 7F+
1	1F 1F 7F 7F
2	0A 0A 7F 7F
3	15 15 7F 7F
4	00 00 00 00

+ Bytes 2 and 3 consist of 7F (hex) pattern to check for stuck on Write Enables at the C180 Decode RAM memory chips. If either byte 0 or byte 1 position memory chips have a stuck on Write Enable, a bit(s) from byte 2 or 3 will be overwritten into the chip and a parity error will occur during the sweep.

Parity Status Format:

See section 12 Description.

Section 12, Subsection 2

This subsection tests IF C170 Decode RAM memory chips with various data patterns, all locations.

Method:

All locations in the IF C170 Decode RAM memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Register 86 and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear.

Repeat the following conditions for each test pattern.

Conditions:

C.0 a. Write all of the IF C170 Decode RAM locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.
c. Clear Errors in processor.

C.1 a. Sweep all locations of IF C170 Decode RAM (locations 00 through FF (hex).
b. Read, format and report IF Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	3F 3F 3F 3F
2	2A 2A 2A 2A
3	15 15 15 15
4	00 00 00 00

NOTE

If any byte from pattern 1 or 2 gets written into the C180 Decode RAM (a hardware fault), a parity error will be reported from the C180 Decode RAM data parity checker.

Parity Status Format:

See Section 12 Description.

Section 12, Subsection 3

This subsection tests the IF C180 Decode RAM read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the IF C180 Decode RAM read data parity checkers. This pattern is written into the Decode RAM memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc). The Decode RAM memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it should not.

Initialization:

- Send Master Clear.

Conditions:

C.0 Write 20 (hex) locations of IF C180 Decode RAM with an incrementing pattern, beginning at location 0 and ending with location 1F (hex); clear Errors in processor; no valid data to report so report 0--0.

Location	Incrementing Pattern (Hexadecimal)
00	00 00 00 00
01	01 01 00 00
02	02 02 00 00
|	|
1E	1E 1E 00 00
1F	1F 1F 00 00

Conditions:

C.1 a. Sweep locations 00 through 1F of IF C180 Decode RAM.
b. Read, format and report IF Parity Status; error if not 0--0.

Parity Status Format:

See section 12 Description.

Section 12, Subsection 4

This subsection tests the IF C170 Decode RAM read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the IF C170 Decode RAM read data parity checkers. This pattern is written into the Decode RAM memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc). The Decode RAM memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it should not.

Initialization:

- Send Master Clear.

Conditions:

C.0 Write 40 (hex) locations of IF C170 Decode RAM with an incrementing pattern, beginning at location 0 and ending with location 3F (hex); clear Errors in processor; no valid data to report so report 0--0.

Location	Incrementing Pattern (Hexadecimal)
00	00 00 00 00
01	01 01 01 01
02	02 02 02 02
|	|
3E	3E 3E 3E 3E
3F	3F 3F 3F 3F

C.1 a. Sweep locations 00 through 3F of IF C170 Decode RAM.
b. Read, format and report IF Parity Status; error if not 0--0.

Parity Status Format:

See section 12 Description.

Section 12, Subsection 5

This subsection tests the IF C180 Decode RAM addressing logic.

Method:

The IF Decode RAMs are built from 1x256 bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Errors to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Set bit 18 of PTM Reg (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in IF C180 Decode RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in IF C180 Decode RAM -- location X+1 if X is not FF (hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear Errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00, Sweep locations 0 through X-1.
b. Read, format and report IF Parity Status; error if not 0--0.

C.4
a. If Test Address X is not FF (hex), Sweep locations X+1 through FF (hex).
b. Read, format and report IF Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report IF Parity Status; error if IF Parity Status is not equal Expected Parity Status.

NOTE

The following steps are executed before repeating the above conditions or exiting the subsection.

- Write Test Address X location with Base pattern.
- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.
- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, without error, clean up memory:
  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear Errors in processor.

Test Patterns (Hexadecimal):

Number	Test Address X	Test Data Pattern	Expected Parity Status
0	00	1F 00 00 00	08
1	00	00 1F 00 00	01
2	00	00 00 1F 00	00
3	00	00 00 00 1F	00
4	55	1F 1F 00 00	09
5	AA	1F 1F 00 00	09
6	FF	1F 1F 00 00	09

Base Patterns (Hexadecimal):

Number	Pattern
0	01 01 00 00

Parity Status Format:

See Section 12 Description.

Section 12, Subsection 6

This subsection tests the IF C170 Decode RAM addressing logic.

Method:

The IF Decode RAMs are built from 1x256 bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

Method (continued):

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Errors to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Set bit 18 of PTM register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in IF C170 Decode RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in IF C170 Decode RAM, location X+1 if X is not FF (hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear Errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00, Sweep locations 0 through X-1.
b. Read, format and report IF Parity Status; error if not 0--0.

C.4
a. If Test Address X is not FF (hex), Sweep locations X+1 through FF (hex).
b. Read, format and report IF Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report IF Parity Status (error if IF Parity Status is not equal; Expected Parity Status).

NOTE

The following steps are executed before repeating the above conditions or exiting the subsection.

- Write Test Address X location with Base pattern.
- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.
- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, without error, clean up memory:
  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear Errors in processor.

Test Patterns (Hexadecimal):

Number	Test Address X	Test Data Pattern	Expected Parity Status
0	00	07 00 00 00	10
1	00	00 07 00 00	02
2	00	00 00 07 00	20
3	00	00 00 00 07	04
4	55	07 07 07 07	36
5	AA	07 07 07 07	36
6	FF	07 07 07 07	36
7	00	07 3C 3C 3C	10
8	00	3C 07 3C 3C	02
9	00	3C 3C 07 3C	20
A	00	3C 3C 3C 07	04

Base Patterns (Hexadecimal):

Number	Pattern
0	38 38 38 38

Parity Status Format:

See Section 12 Description.

Section 12, Subsection 7

This subsection tests the addressing circuitry internal to the IF C180 Decode RAM memory chips.

Method:

The IF Decode RAMs are built from 1 by 256 bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error ocurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping IF Decode RAMs.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1 Report Test Pattern data in Condition Result bytes 4 through 7; informational.

Conditions:

C.2 Report Test Address X in Condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in IF C180 Decode RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
b. Read, format and report Parity Status; error if not 0--0.

If Repeat Iteration set, repeat C.3.

C.4
a. Set bit 18 of PTM register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM register.
d. Nothing valid to report so report 0--0.

- If Test Address X is not equal to last pattern value, increment Test Address X and repeat C.2 through C.4.

Exit subsection if no more base patterns.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	1F 1F 00 00
01	|
02	|
incrementing address	Same Data Pattern |
FD	|
FE	1F 1F 00 00

Base Patterns (Hexadecimal):

0	01 01 00 00
1	10 10 00 00

Parity Status Format:

See Section 12 Description.

Section 12, Subsection 8

This subsection tests the addressing circuitry internal to the IF C170 Decode RAM memory chips.

Method:

The IF Decode RAMs are built from 1 by 256 bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping IF Decode RAMs.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in condition Result bytes 4 through 7; informational.

C.1 Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2 Report Test Address X in Condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in IF C170 Decode RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

Conditions:

C.4
If C.3 did not fail,
a. Set bit 18 of PTM register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM register.
d. Nothing valid to report so report 0--0.

Else, (C.3 did fail) nothing valid to report so report 0--0. If Repeat Iteration set, repeat C.3 and C.4.

If C.3 did not fail and Test Address X is not equal to last pattern value, increment Test Address X and repeat C.2 through C.4.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	07 07 07 07
01	|
02	|
incrementing address	Same Data Pattern
FD	|
FE	07 07 07 07

Base Patterns (Hexadecimal):

0 38 38 38 38

Parity Status Format:

See Section 12 Description.

SECTION 13 - TEST AC SOFT CONTROL 1 RAM MEMORIES

This section completely verifies the data holding and addressing capabilities of the AC Soft Control 1 memory logic.

Subsection 0 - Test AC Soft Control 1 memory chips with various data patterns, all locations.

Subsection 1 - Test AC Soft Control 1 RAM read data parity checkers.

Subsection 2 - Test AC Soft Control 1 addressing logic.

Subsection 3 - Test the addressing circuitry internal to the AC Soft Control 1 memory chips.

Hardware Tested:

- The AC Soft Control 1 memory chips.
- The AC Soft Control 1 memory chip read data parity checkers.
- The AC Soft Control 1 addressing logic.

Method:

See the individual subsections Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	AC SC 1 (loc A2G1)	80	46
62	AC SC 2 (loc A2E1)	80	45
61	ALN SC PAK (loc C2K1)	80	44

Section 13, Subsection 0

This subsection tests AC Soft Control 1 memory chips with various data patterns, all locations.

Method:

All locations in the AC Soft Control 1 RAM memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the AC Soft Control 1 memory locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of AC Soft Control 1 memory (locations 00 through 3F(hex)).
b. Read, format and report SCM Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	FF FF FF FF
2	AA AA AA AA
3	55 55 55 55
4	7F 7F 7F 7F
5	80 80 80 80
6	7F 00 7F 00
7	00 7F 00 7F
8	00 00 00 00

Parity Status Format:

See Section 13 Description.

Section 13, Subsection 1

This subsection tests AC Soft Control 1 RAM read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the AC Soft Control PAK read data parity checkers. This pattern is written into the AC Soft Control PAK memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc.). The AC Soft Control PAK memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it shouldn't.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0 a. Write all of AC Soft Control 1 memory (40(hex) locations) with an incrementing pattern 00 through 3F(hex):

Location	Incrementing Pattern (Hexadecimal)
00	00 00 00 00
01	01 01 01 01
02	02 02 02 02
|	|
3E	3E 3E 3E 3E
3F	3F 3F 3F 3F

b. Sweep all locations of AC Soft Control 1 memory (40(hex) locations).
c. Read, format and report SCM Parity Status; error if not 0.

C.1 a. Clear errors in processor.
b. Write all of AC Soft Control 1 memory (40(hex) locations) with an incrementing pattern 40(hex) through 7F:

Location	Incrementing Pattern (Hexadecimal)
00	40 40 40 40
01	41 41 41 41
02	42 42 42 42
|	|
3E	7E 7E 7E 7E
3F	7F 7F 7F 7F

Conditions (C.1 continued):

c. Sweep all locations of AC Soft Control 1 memory (40(hex) locations).

d. Read, format and report SCM Parity Status; error if not 0.

C.2 a. Clear Errors in processor.

b. Write all of AC Soft Control 1 memory (40(hex) locations) with an incrementing pattern 80(hex) through BF(hex):

Location	Incrementing Pattern (Hexadecimal)
00	80 80 80 80
01	81 81 81 81
02	82 82 82 82
|	|
3E	BE BE BE BE
3F	BF BF BF BF

c. Sweep all locations of AC Soft Control 1 memory (40(hex) locations).
d. Read, format and report SCM Parity Status; error if not 0.

C.3 a. Clear errors in processor.
b. Write all of AC Soft Control 1 memory (40(hex) locations) with an incrementing pattern C0(hex) through FF(hex):

Location	Incrementing Pattern (Hexadecimal)
00	C0 C0 C0 C0
01	C1 C1 C1 C1
02	C2 C2 C2 C2
|	|
3E	FE FE FE FE
3F	FF FF FF FF

c. Sweep all locations of AC Soft Control 1 memory (40(hex) locations).
d. Read, format and report SCM Parity Status; error if not 0.

Parity Status Format:

See Section 13 Description.

Section 13, Subsection 2

This subsection tests AC Soft Control 1 addressing logic.

Method:

The AC Soft Control PAK RAMs are built from 4 by 16 bit memory chips, one chip for each 4 bit positions in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization

- Send Master Clear and Clear Errors to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

Conditions:

C.1
a. Set bit 18 of PTM register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in AC Soft Control PAK RAM, (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in AC Soft Control PAK RAM, location X+1 if X is not 3F(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00, Sweep locations 0 through X-1.
b. Read, format and report SCM Parity Status; error if not 0--0.

C.4
a. If Test Address X is not 3F(hex), Sweep locations X+1 through 3F(hex).
b. Read, format and report SCM Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report SCM Parity Status; error if SCM Parity Status is not equal Expected Parity Status.

NOTE

The following steps are executed before repeating the above conditions or exiting the subsection.

- Write Test Address X location with Base pattern.
- If Repeat Iteration set, go to C.1.
- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.
- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, without error, clean up memory:
  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear Errors in processor.

Test Patterns (Hexadecimal):

No.	Test Address X	Test Data Pattern	Expected Parity Status
0	00	B0 B1 B1 B1	01
1	00	B1 B0 B1 B1	01
2	00	B1 B1 B0 B1	01
3	00	B1 B1 B1 B0	01
4	15	B0 B1 B1 B1	01
5	15	B1 B0 B1 B1	01
6	15	B1 B1 B0 B1	01
7	15	B1 B1 B1 B0	01
8	2A	B0 B1 B1 B1	01
9	2A	B1 B0 B1 B1	01
A	2A	B1 B1 B0 B1	01
B	2A	B1 B1 B1 B0	01
C	3F	B0 B1 B1 B1	01
D	3F	B1 B0 B1 B1	01
E	3F	B1 B1 B0 B1	01
F	3F	B1 B1 B1 B0	01
10	00	B0 00 00 00	01
11	00	00 B0 00 00	01
12	00	00 00 B0 00	01
13	00	00 00 00 B0	01
14	15	B0 00 00 00	01
15	15	00 B0 00 00	01
16	15	00 00 B0 00	01
17	15	00 00 00 B0	01
18	2A	B0 00 00 00	01
19	2A	00 B0 00 00	01
1A	2A	00 00 B0 00	01
1B	2A	00 00 00 B0	01
1C	3F	B0 00 00 00	01
1D	3F	00 B0 00 00	01
1E	3F	00 00 B0 00	01
1F	3F	00 00 00 B0	01

Base Patterns (Hexadecimal):

No.	Pattern
0	57 57 57 57

Parity Status Format:

See Section 13 Description.

Section 13, Subsection 3

This subsection tests the addressing circuitry internal to the AC Soft Control 1 memory chips.

Method:

The AC Soft Control PAK RAMs are built from 4 by 16 bit memory chips, one chip for each 4 bit positions in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward and reverse March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping AC SC memories.

c. Write location just swept with special test pattern containing bad parity.

   Steps b and c are repeated using incrementing test addresses (00 to 3F(hex)) until a parity error is sensed, or all locations have been swept and written in a forward direction.

d. Write the entire memory with the base pattern.

e. Sweep two locations (Test Address X and X-1) and verify that no data parity error occurred.

f. Write the location just swept (X) with special test pattern containing bad parity.

   Steps e and f are repeated using decrementing test addresses (3F(hex) to 00) until a parity error is sensed or all locations have been swept and written in a reverse direction.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Initialize Test Address X to 00.
b. Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2
Report Test Address X in condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in AC Soft Control 1 RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

C.4
If C.3 did not fail,
a. Set bit 18 of PTM register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM register.
d. Nothing valid to report so report 0--0.

Else, (C.3 did fail) nothing valid to report so report 0--0.

If C.3 did not fail and Test Address X is not equal to 3F(hex), increment Test Address X and repeat C.2 through C.4.

C.5
a. Write all locations in the memory with Base Pattern.
b. Initialize Test Address X to 3F(hex).
c. Nothing valid to report, so report 0--0.

C.6
Report Test Address X in Condition Result byte 7; informational.

C.7
a. Sweep Test Address X location in AC Soft Control 1 RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

C.8
If C.7 did not fail
a. Set bit 18 of PTM register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM register.
d. Nothing valid to report, so report 0--0.

Else, (C.7 did fail) nothing valid to report so report 0--0.

If C.7 did not fail and Test Address X is not equal to 00, decrement Test Address X and repeat C.6 through C.8.

Exit subsection if no more base patterns.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	B0 B0 B0 B0
01	|
02	|
incrementing address	Same data pattern
3E	|
3F	B0 B0 B0 B0

Base Patterns (Hexadecimal):

0	57 57 57 57
1	CE CE CE CE

Parity Status Format:

See Section 13 Description.

SECTION 14 - TEST AC SOFT CONTROL 2 RAM MEMORIES

This section completely verifies the data holding and addressing capabilities of the AC Soft Control 2 memory logic.

Subsection 0 - Test AC Soft Control 2 memory chips with various data patterns, all locations.

Subsection 1 - Test AC Soft Control 2 RAM read data parity checkers.

Subsection 2 - Test AC Soft Control 2 addressing logic.

Subsection 3 - Test the addressing circuitry internal to the AC Soft Control 2 memory chips.

Hardware Tested:

- The AC Soft Control 2 memory chips.
- The AC Soft Control 2 memory chip read data parity checkers.
- The AC Soft Control 2 addressing logic.

Method:

See the individual subsections Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	AC SC 1 (loc A2G1)	80	46
62	AC SC 2 (loc A2E1)	80	45
61	ALN SC PAK (loc C2K1)	80	44

Section 14, Subsection 0

This subsection tests AC Soft Control 2 memory chips with various data patterns, all locations.

Method:

All locations in the AC Soft Control 2 RAM memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the AC Soft Control 2 memory locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of AC Soft Control 2 memory (locations 00 through FF(hex)).
b. Read, format and report SCM Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	FF FF FF FF
2	AA AA AA AA
3	55 55 55 55
4	7F 7F 7F 7F
5	80 80 80 80
6	7F 00 7F 00
7	00 7F 00 7F
8	00 00 00 00

Parity Status Format:

See Section 14 Description.

Section 14, Subsection 1

This subsection tests AC Soft Control 2 RAM read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the AC Soft Control 2 read data parity checkers. This pattern is written into the AC Soft Control 2 memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc.). The AC Soft Control 2 memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it shouldn't.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0 a. Write all of AC Soft Control 2 memory (40(hex) locations) with an incrementing pattern 00 through 3F(hex):

Location	Incrementing Pattern (Hexadecimal)
00	00 00 00 00
01	01 01 01 01
02	02 02 02 02
|	|
3E	3E 3E 3E 3E
3F	3F 3F 3F 3F

b. Sweep all locations of AC Soft Control 2 memory (40(hex) locations).
c. Read, format and report SCM Parity Status; error if not 0.

C.1 a. Clear errors in processor.
b. Write all of AC Soft Control 2 memory (40(hex) locations) with an incrementing pattern 40(hex) through 7F(hex):

Location	Incrementing Pattern (Hexadecimal)
00	40 40 40 40
01	41 41 41 41
02	42 42 42 42
|	|
3E	7E 7E 7E 7E
3F	7F 7F 7F 7F

Conditions (C.1 continued):

c. Sweep all locations of AC Soft Control 2 memory (40(hex) locations).

d. Read, format and report SCM Parity Status; error if not 0.

C.2 a. Clear errors in processor.

b. Write all of AC Soft Control 2 memory (40(hex) locations) with an incrementing pattern 80 through BF(hex):

Location	Incrementing Pattern (Hexadecimal)
00	80 80 80 80
01	81 81 81 81
02	82 82 82 82
|	|
3E	BE BE BE BE
3F	BF BF BF BF

c. Sweep all locations of AC Soft Control 2 memory (40(hex) locations).
d. Read, format and report SCM Parity Status; error if not 0.

C.3 a. Clear errors in processor.
b. Write all of AC Soft Control 2 memory (40(hex) locations) with an incrementing pattern C0(hex) through FF(hex):

Location	Incrementing Pattern (Hexadecimal)
00	C0 C0 C0 C0
01	C1 C1 C1 C1
02	C2 C2 C2 C2
|	|
3E	FE FE FE FE
3F	FF FF FF FF

c. Sweep all locations of AC Soft Control 2 memory (40(hex) locations).
d. Read, format and report SCM Parity Status; error if not 0.

Parity Status Format:

See Section 14 Description.

Section 14, Subsection 2

This subsection tests AC Soft Control 2 addressing logic.

Method:

The AC Soft Control 2 RAMs are built from 4 by 16-bit memory chips, one chip for each four bit positions in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).
b. Write one location (Test Address X) with a special test pattern containing bad parity.
c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.
d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address used to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

n order to thoroughly test some memories, more than one base pattern is ometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Errors to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in AC Soft Control 2 RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Report Test Address X; informational.

Conditions:

C.2
a. Sweep one good location in AC Soft Control 2 RAM, location X+1 if X is not 3F(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00, Sweep locations 0 through X-1.
b. Read, format and report SCM Parity Status; error if not 0--0.

C.4
a. If Test Address X is not 3F(hex), Sweep locations X+1 through 3F(hex).
b. Read, format and report SCM Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report SCM Parity Status; error if SCM Parity Status is not equal Expected Parity Status.

NOTE

The following steps are executed before repeating the above conditions or exiting the subsection.

- Write Test Address X location with Base pattern.
- If Repeat Iteration set, go to C.1.
- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.
- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, without error, clean up memory:
  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear errors in processor.

Test Patterns (Hexadecimal):

Number	Test Address X	Test Data Pattern	Expected Parity Status
0	00	B0 B1 B1 B1	02
1	00	B1 B0 B1 B1	02
2	00	B1 B1 B0 B1	02
3	00	B1 B1 B1 B0	02
4	15	B0 B1 B1 B1	02
5	15	B1 B0 B1 B1	02
6	15	B1 B1 B0 B1	02
7	15	B1 B1 B1 B0	02
8	2A	B0 B1 B1 B1	02
9	2A	B1 B0 B1 B1	02
A	2A	B1 B1 B0 B1	02
B	2A	B1 B1 B1 B0	02
C	3F	B0 B1 B1 B1	02
D	3F	B1 B0 B1 B1	02
E	3F	B1 B1 B0 B1	02
F	3F	B1 B1 B1 B0	02
10	00	B0 00 00 00	02
11	00	00 B0 00 00	02
12	00	00 00 B0 00	02
13	00	00 00 00 B0	02
14	15	B0 00 00 00	02
15	15	00 B0 00 00	02
16	15	00 00 B0 00	02
17	15	00 00 00 B0	02
18	2A	B0 00 00 00	02
19	2A	00 B0 00 00	02
1A	2A	00 00 B0 00	02
1B	2A	00 00 00 B0	02
1C	3F	B0 00 00 00	02
1D	3F	00 B0 00 00	02
1E	3F	00 00 B0 00	02
1F	3F	00 00 00 B0	02

Base Patterns (Hexadecimal):

0 57 57 57 57

Parity Status Format:

See Section 14 Description.

Section 14, Subsection 3

This subsection tests the addressing circuitry internal to the AC Soft Control 2 memory chips.

Method:

The AC SC RAMs are built from 4 by 16-bit memory chips, one chip for each 4 bit positions in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward and reverse March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping AC SC memories.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses (00 to 3F(hex)) until a parity error is sensed, or all locations have been swept and written in a forward direction.

d. Write the entire memory with the base pattern.

e. Sweep two locations (Test Address X and X-1) and verify that no data parity error occurred.

f. Write the location just swept (X) with special test pattern containing bad parity.

Steps e and f are repeated using decrementing test addresses (3F(hex) to 00) until a parity error is sensed or all locations have been swept and written in a reverse direction.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Initialize Test Address X to 00.
b. Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2 Report Test Address X in Condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in AC Soft Control 2 RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

C.4 If C.3 did not fail.
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report so report 0--0.

Else, (C.3 did fail) nothing valid to report so report 0--0.

If C.3 did not fail and Test Address X is not equal to 3F(hex) increment Test Address X and repeat C.2 through C.4.

C.5
a. Write all locations in the memory with Base Pattern.
b. Initialize Test Address X to 3F(hex).
c. Nothing valid to report, so report 0--0.

C.6 Report Test Address X in Condition Result byte 7; informational.

C.7
a. Sweep Test Address X location in AC Soft Control 2 RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

C.8 If C.7 did not fail
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report, so report 0--0.

Else, (C.7 did fail) nothing valid to report so report 0--0.

If C.7 did not fail and Test Address X is not equal to 00, decrement Test Address X and repeat C.6 through C.8.

Exit subsection if no more base patterns.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	B0 B0 B0 B0
01	|
02	|
incrementing address	Same data pattern
3E	|
3F	B0 B0 B0 B0

Base Patterns (Hexadecimal):

0	57 57 57 57
1	CE CE CE CE

Parity Status Format:

See Section 14 Description.

SECTION 15 - TEST ALN SC RAM MEMORIES

This section completely verifies the data holding and addressing capabilities of the ALN SC memory logic.

Subsection 0 - Tests ALN SC memory chips with various data patterns, all locations.

Subsection 1 - Tests ALN SC RAM read data parity checkers.

Subsection 2 - Tests ALN SC addressing logic.

Subsection 3 - Tests the addressing circuitry internal to the ALN SC memory chips.

Hardware Tested:

- The ALN SC memory chips.
- The ALN SC memory chip read data parity checkers.
- The ALN SC addressing logic; including Address Mux A and B (MAC leg), Address Registers A and B, Chip Select Decode logic A and B, Write Byte Selects, and addressing circuitry internal to each memory chip (4 X 16 bit chips).

Method:

See the individual subsection Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	AC SC 1 (loc A2G1)	80	46
62	AC SC 2 (loc A2E1)	80	45
61	ALN SC (loc C2K1)	80	44

Section 15, Subsection 0

This subsection tests ALN SC memory chips with various data patterns, all locations.

Method:

All locations in the ALN SC RAM memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the ALN SC memory locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of ALN SC memory (locations 00 through 3F(hex)).
b. Read, format and report SCM Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	FF FF FF FF
2	AA AA AA AA
3	55 55 55 55
4	7F 7F 7F 7F
5	80 80 80 80
6	7F 00 7F 00
7	00 7F 00 7F
8	00 00 00 00

Parity Status Format:

See Section 15 Description.

Section 15, Subsection 1

This subsection tests ALN SC RAM read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the ALN SC read data parity checkers. This pattern is written into the ALN SC memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc.). The ALN SC memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it shouldn't.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0 a. Write all of ALN SC memory (40(hex) locations) with an incrementing pattern 00 through 3F(hex):

Location	Incrementing Pattern (Hexadecimal)
00	00 00 00 00
01	01 01 01 01
02	02 02 02 02
|	|
3E	3E 3E 3E 3E
3F	3F 3F 3F 3F

b. Sweep all locations of ALN SC memory (40(hex) locations).
c. Read, format and report SCM Parity Status; error if not 0.

C.1 a. Clear errors in processor.
b. Write all of ALN SC memory (40(hex) locations) with an incrementing pattern 40(hex) through 7F(hex):

Location	Incrementing Pattern (Hexadecimal)
00	40 40 40 40
01	41 41 41 41
02	42 42 42 42
|	|
3E	7E 7E 7E 7E
3F	7F 7F 7F 7F

Conditions (C.1 continued):

c. Sweep all locations of ALN SC memory (40(hex) locations).
d. Read, format and report SC memory Parity Status; error if not 0.

C.2 a. Clear errors in processor.
b. Write all of ALN SC memory (40(hex) locations) with an incrementing pattern 80(hex) through BF(hex):

Location	Incrementing Pattern (Hexadecimal)
00	80 80 80 80
01	81 81 81 81
02	82 82 82 82
|	|
3E	BE BE BE BE
3F	BF BF BF BF

c. Sweep all locations of ALN SC memory (40(hex) locations).
d. Read, format and report SC memory Parity Status; error if not 0.

C.3 a. Clear errors in processor.
b. Write all of ALN SC memory (40(hex) locations) with an incrementing pattern C0(hex) through FF(hex):

Location	Incrementing Pattern (Hexadecimal)
00	C0 C0 C0 C0
01	C1 C1 C1 C1
02	C2 C2 C2 C2
|	|
3E	FE FE FE FE
3F	FF FF FF FF

c. Sweep all locations of ALN SC memory (40(hex) locations).
d. Read, format and report SC memory Parity Status; error if not 0.

Parity Status Format:

See Section 15 Description.

Section 15, Subsection 2

This subsection tests ALN SC addressing logic.

Method:

The ALN SC RAMs are built from 4 by 16-bit memory chips, one chip for each 4-bit positions in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in ALN SC RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Report Test Address X; informational.

Conditions:

C.2
a. Sweep one good location in ALN SC RAM, location X+1 if X is not 3F(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00 sweep locations 0 through X-1.
b. Read, format and report SCM Parity Status; error if not 0--0.

C.4
a. If Test Address X is not 3F(hex), sweep locations X+1 through 3F(hex).
b. Read, format and report SCM Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report SCM Parity Status; error if SCM Parity Status is not equal Expected Parity Status.

NOTE

The following steps are executed before repeating the above conditions for exiting the subsection.

- Write Test Address X location with Base pattern.
- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.
- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, without error, clean up memory:
  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear errors in processor.

Test Patterns (Hexadecimal):

No.	Test Address X	Test Data Pattern	Expected Parity Status
0	00	B0 B1 B1 B1	04
1	00	B1 B0 B1 B1	|
2	00	B1 B1 B0 B1	|
3	00	B1 B1 B1 B0	|
4	15	B0 B1 B1 B1	|
5	15	B1 B0 B1 B1	|
6	15	B1 B1 B0 B1	|
7	15	B1 B1 B1 B0	Same Parity
8	2A	B0 B1 B1 B1	Status
9	2A	B1 B0 B1 B1	|
A	2A	B1 B1 B0 B1	|
B	2A	B1 B1 B1 B0	|
C	3F	B0 B1 B1 B1	|
D	3F	B1 B0 B1 B1	|
E	3F	B1 B1 B0 B1	|
F	3F	B1 B1 B1 B0	|
10	00	B0 00 00 00	|
11	00	00 B0 00 00	|
12	00	00 00 B0 00	|
13	00	00 00 00 B0	|
14	15	B0 00 00 00	|
15	15	00 B0 00 00	|
16	15	00 00 B0 00	|
17	15	00 00 00 B0	|
18	2A	B0 00 00 00	|
19	2A	00 B0 00 00	|
1A	2A	00 00 B0 00	|
1B	2A	00 00 00 B0	|
1C	3F	B0 00 00 00	|
1D	3F	00 B0 00 00	|
1E	3F	00 00 B0 00	|
1F	3F	00 00 00 B0	04

Base Patterns (Hexadecimal):

0 57 57 57 57

Parity Status Format:

See Section 15 Description.

Section 15, Subsection 3

This subsection tests the addressing circuitry internal to the ALN SC memory chips.

Method:

The ALN SC RAMs are built from 4 by 16-bit memory chips, one chip for each 4 bit positions in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward and reverse March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping the ALN SC memory.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses (00 to 3F(hex)) until a parity error is sensed, or all locations have been swept and written in a forward direction.

d. Write the entire memory with the base pattern.

e. Sweep two locations (Test Address X and X-1) and verify that no data parity error occurred.

f. Write the location just swept (X) with special test pattern containing bad parity.

Steps e and f are repeated using Decrementing Test Addresses (3F(hex) to 00) until a parity error is sensed or all locations have been swept and written in a reverse direction.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1 Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2 Report Test Address X in Condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in ALN SC RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

C.4 If C.3 did not fail,
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report so report 0--0.

Else, (C.3 did fail) nothing valid to report so report 0--0.

If C.3 did not fail and Test Address X is not equal to 3F(hex), increment Test Address X and repeat C.2 through C.4.

C.5
a. Write all locations in the memory with Base Pattern.
b. Initialize Test Address X to 3F(hex).
c. Nothing valid to report, so report 0--0.

C.6 Report Test Address X in Condition Result byte 7; informational.

C.7
a. Sweep Test Address X location in ALN SC RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

C.8 If C.7 did not fail.
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report, so report 0--0.

Else, (C.7 did fail) nothing valid to report so report 0--0.

If C.7 did not fail and Test Address X is not equal to 00, decrement Test Address X and repeat C.6 through C.8.

Exit subsection if no more patterns.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	B0 B0 B0 B0
01	|
02	|
incrementing address	Same data pattern |
3E	|
3F	B0 B0 B0 B0

Base Patterns (Hexadecimal):

0	57 57 57 57
1	CE CE CE CE

Parity Status Format:

See Section 15 Description.

SECTION 16 - TEST BDP PORT A MEMORY BOARD

This section completely verifies the data holding and addressing capabilities of the BDP Port A board's memory logic.

Subsection 0 - Tests BDP Port A board's memory chips with various data patterns, all locations.

Subsection 1 - Tests BDP Port A board's memory read data parity checkers.

Subsection 2 - Tests BDP Port A board's addressing logic.

Subsection 3 - Tests the addressing circuitry internal to the BDP Port A Board's memory chips.

Hardware Tested:

- The BDP Port A board's memory chips.
- The BDP Port A board's memory chip read data parity checkers.
- The BDP Port A board's addressing logic.

Method:

See the individual subsection Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At		PFS Register No.	Bit
63	BDP RAM PAK 0-B Data	(loc A3A3)	81	11
62	BDP RAM PAK 0-B Address	(loc A3A2)	81	10
61	BDP RAM PAK 0-A Data	(loc A3A2)	81	9
60	BDP RAM PAK 0-A Address	(loc A3A2)	81	8
59	BDP Port B Stage 2 Rgtr	(loc A3C1)	81	5
58	BDP Port A Stage 2 Rgtr	(loc A3B1)	81	4
57	BDP RAM PAK 1 Data	(loc A3B2)	81	3
56	BDP RAM PAK 1 Address	(loc A3B2)	81	2
55	BDP Port A RAM Address	(loc A3B1)	80	55
54	BDP Port B RAM Address	(loc A3C1)	80	54

Section 16, Subsection 0

This subsection tests BDP Port A board's memory chips with various data patterns, all locations.

Method:

All locations in the BDP Port A board's memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the BDP Port A memory locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of BDP Port A memory (locations 00 through FF(hex)).
b. Read, format and report BDP Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	00 FF FE FF
2	00 AA AA AA
3	00 55 54 55
4	00 7F 7E 7F
5	00 80 80 80
6	00 00 00 00

NOTE

Bit 23 of BDP Port A board memory data must always be 0 in order to provide proper parity in the memory.

Parity Status Format:

See Section 16 Description.

Section 16, Subsection 1

This subsection tests BDP Port A board's memory read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the BDP Port A RAM read data parity checkers. This pattern is written into the RAM memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc). The RAM memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it shouldn't.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0 Write all locations of the BDP Port A memory with an incrementing pattern, beginning at location 0 and ending with location FF(hex); no valid data to report so report 0--0.

Location	Incrementing Pattern (Hexadecimal)	
00	00 00 00 00	
01	00 01 00 01	
02	00 02 02 02	bit 23
|	|	must
FE	00 FE FE FE	always be
FF	00 FF FE FF	clear.

C.1 a. Sweep locations 00 through FF of BDP Port A memory.
b. Read, format and report BDP Parity Status; error if not 0--0.

Parity Status Format:

See section 16 Description.

Section 16, Subsection 2

This subsection tests BDP Port A board's addressing logic.

Method:

The BDP Port A RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

Conditions:

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in BDP Port A RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in BDP Port A RAM, location X+1 if X is not FF(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00, Sweep locations 0 through X-1.
b. Read, format and report BDP Parity Status; error if not 0--0.

C.4
a. If Test Address X is not FF(hex), Sweep locations X+1 through FF(hex).
b. Read, format and report BDP Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report BDP Parity Status; error if BDP Parity Status is not equal Expected Parity Status.

- Write Test Address X location with Base pattern.
- If Repeat Iteration set, go to C.1.
- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.
- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, clean up memory:
  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear errors in processor.

Test Patterns (Hexadecimal):

Number	Test Address X	Test Data Pattern	Expected Parity Status
0	00	7F 00 00 00	00
1	00	00 7F 00 00	20
2	00	C0 C0 3E C0	20
3	00	00 00 00 7F	20
Number	Test Address X	Test Data Pattern	Expected Parity Status
4	00	80 00 00 00	00
5	00	00 80 00 00	20
6	00	81 81 3E 81	20
7	00	00 00 00 80	20
8	55	00 7F 3E 7F	20
9	AA	00 7F 3E 7F	20
A	FF	00 7F 3E 7F	20

NOTE

Bit 23 of BDP Port A board memory data must always be 0 in order to provide proper parity in the memory.

Base Patterns (Hexadecimal):

0 00 80 80 80

Parity Status Format:

See Section 16 Description.

Section 16, Subsection 3

This subsection tests the addressing circuitry internal to the BDP Port A board's memory chips.

Method:

The BDP Port A RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping BDP memories.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Conditon Result bytes 4 through 7; informational.

C.1 Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2 Report Test Address X in Condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in BDP Port A RAM
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

If Repeat Iteration set, repeat C.3.

C.4
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report so report 0--0.

If Test Address X is not equal to FF(hex), increment Test Address X and repeat C.2 through C.4

Exit subsection if no more base patterns.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	00 7F 3E 7F
01	|
02	|
incrementing address	Same data pattern
FE	|
FF	00 7F 3E 7F

Base Patterns (Hexadecimal):

0 00 80 80 80
1 00 80 40 80

Parity Status Format:

See Section 16 Description.

SECTION 17 - TEST BDP PORT B MEMORY BOARD

This section completely verifies the data holding and addressing capabilities of the BDP Port B board's memory logic.

Subsection 0 - Tests BDP Port B board's memory chips with various data patterns, all locations.

Subsection 1 - Tests BDP Port B board's memory read data parity checkers.

Subsection 2 - Tests BDP Port B board's addressing logic.

Subsection 3 - Tests the addressing circuitry internal to the BDP Port B Board's memory chips.

Hardware Tested:

- The BDP Port B board's memory chips.
- The BDP Port B board's memory chip read data parity checkers.
- The BDP Port B board's addressing logic.

Method:

See the individual subsections Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At		PFS Register No.	Bit
63	BDP RAM PAK 0-B Data	(loc A3A3)	81	11
62	BDP RAM PAK 0-B Address	(loc A3A2)	81	10
61	BDP RAM PAK 0-A Data	(loc A3A2)	81	9
60	BDP RAM PAK 0-A Address	(loc A3A2)	81	8
59	BDP Port B Stage 2 Rgtr	(loc A3C1)	81	5
58	BDP Port A Stage 2 Rgtr	(loc A3B1)	81	4
57	BDP RAM PAK 1 Data	(loc A3B2)	81	3
56	BDP RAM PAK 1 Address	(loc A3B2)	81	2
55	BDP Port A RAM Address	(loc A3B1)	80	55
54	BDP Port B RAM Address	(loc A3C1)	80	54

Section 17, Subsection 0

This subsection tests BDP Port B board's memory chips with various data patterns, all locations.

Method:

All locations in the BDP Port B board's memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the BDP Port B RAM locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of BDP Port B RAM (locations 00 through FF(hex)).
b. Read, format and report BDP Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	00 FF FE FF
2	00 AA AA AA
3	00 55 54 55
4	00 7F 7E 7F
5	00 80 80 80
6	00 00 00 00

NOTE

Bit 23 of BDP Port B board memory data must always be 0 in order to provide proper parity in the memory.

Parity Status Format:

See Section 17 Description.

Section 17, Subsection 1

This subsection tests BDP Port B board's memory read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the BDP Port B RAM read data parity checkers. This pattern is written into the RAM memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc). The RAM memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it should not occur.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0 Write all locations of the BDP Port B memory with an incrementing pattern, beginning at location 0 and ending with location FF(hex); no valid data to report so report 0--0.

Location	Incrementing Pattern (Hexadecimal)	
00	00 00 00 00	
01	00 01 00 01	
02	00 02 02 02	Bit 23
|	|	must always
FE	00 FE FE FE	be clear
FF	00 FF FE FF	

Conditions:

C.1 a. Sweep locations 00 through FF of BDP Port B memory.
b. Read, format and report BDP Parity Status; error if not 0--0.

Parity Status Format:

See section 17 Description.

Section 17, Subsection 2

This subsection tests BDP Port B board's addressing logic.

Method:

The BDP Port B RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in BDP Port B RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in BDP Port B RAM, location X+1 if X is not FF(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear Errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00 Sweep locations 0 through X-1.
b. Read, format and report BDP Parity Status; error if not 0--0.

C.4
a. If Test Address X is not FF(hex), Sweep locations X+1 through FF(hex).
b. Read, format and report BDP Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report BDP Parity Status; error if BDP Parity Status is not equal Expected Parity Status.

- Write Test Address X location with Base pattern.

- If Repeat Iteration set, go to C.1.

- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.

- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, clean up memory:

  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear errors in processor.

Test Patterns (Hexadecimal):

Number	Test Address X	Test Data Pattern	Expected Parity Status
0	00	7F 00 00 00	00
1	00	00 7F 00 00	10
2	00	C0 C0 3E C0	10
3	00	00 00 00 7F	10
4	00	80 00 00 00	00
5	00	00 80 00 00	10
6	00	81 81 3E 81	10
7	00	00 00 00 80	10
8	55	00 7F 3E 7F	10
9	AA	00 7F 3E 7F	10
A	FF	00 7F 3E 7F	10

Note

Bit 23 of BDP Port B board memory data must always be 0 in order to provide proper parity in the memory.

Base Patterns (Hexadecimal):

0 00 80 80 80

Parity Status Format:

See Section 17 Description.

Section 17, Subsection 3

This subsection tests the addressing circuitry internal to the BDP Port B board's memory chips.

Method:

The BDP Port B RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping BDP memories.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0 a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1 Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2 Report Test Address X in Condition Result byte 7; informational.

C.3 a. Sweep Test Address X location in BDP Port B RAM
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

If Repeat Iteration set, repeat C.3.

C.4 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report so report 0--0.

Conditions:

If Test Address X is not equal to FF(hex), increment Test Address X and repeat C.2 through C.4.

Exit subsection if no more base patterns.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	00 7F 3E 7F
01	|
02	|
incrementing address	Same data pattern
FE	|
FF	00 7F 3E 7F

Base Patterns (Hexadecimal):

0	00 80 80 80
1	00 80 40 80

Parity Status Format:

See Section 17 Description.

SECTION 18 - TEST BDP RAM PAK 0 EBCDIC MEMORY

This section completely verifies the data holding and addressing capabilities of the BDP RAM PAK 0 EBCDIC memory logic (memory A).

Subsection 0 - Tests BDP RAM PAK 0 EBCDIC memory chips with various data patterns, all locations.

Subsection 1 - Tests BDP RAM PAK 0 EBCDIC memory read data parity checkers.

Subsection 2 - Tests BDP RAM PAK 0 EBCDIC memory addressing logic.

Subsection 3 - Tests the addressing circuitry internal to the BDP RAM PAK 0 EBCDIC memory chips.

Hardware Tested:

- The BDP RAM PAK 0 EBCDIC memory chips.
- The BDP RAM PAK 0 EBCDIC memory chip read data parity checkers.
- The BDP RAM PAK 0 EBCDIC memory addressing logic.

Method:

See the individual subsections Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At		PFS Register No.	PFS Register Bit
63	BDP RAM PAK 0 - B Data	(loc A3A2)	81	11
62	BDP RAM PAK 0 - B Address	(loc A3A2)	81	10
61	BDP RAM PAK 0 - A Data	(loc A3A2)	81	9
60	BDP RAM PAK 0 - A Address	(loc A3A2)	81	8
59	BDP Port B Stage 2 Rgtr	(loc A3C1)	81	5
58	BDP Port A Stage 2 Rgtr	(loc A3B1)	81	4
57	BDP RAM PAK 1 Data	(loc A3B2)	81	3
56	BDP RAM PAK 1 Address	(loc A3B2)	81	2
55	BDP Port A RAM Address	(loc A3B1)	80	55
54	BDP Port B RAM Address	(loc A3C1)	80	54

Section 18, Subsection 0

This subsection tests BDP RAM PAK 0 EBCDIC memory chips with various data patterns, all locations.

Method:

All locations in the BDP PAK 0 EBCDIC memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the BDP PAK 0 EBCDIC RAM locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of BDP PAK 0 EBCDIC RAM (locations 00 through FF(hex)).
b. Read, format and report BDP Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	00 00 03 FF
2	00 00 02 AA
3	00 00 01 55
4	00 00 03 7F
5	00 00 00 80
6	00 00 00 00

Parity Status Format:

See Section 18 Description

Section 18, Subsection 1

This subsection tests BDP RAM PAK 0 EBCDIC memory read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the BDP PAK 0 EBCDIC RAM read data parity checkers. This pattern is written into the RAM memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc). The RAM memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it shouldn't.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0 Write all locations of BDP PAK 0 EBCDIC RAM with an incrementing pattern, beginning at location 0 and ending with location FF(hex); no valid data to report so report 0--0.

Location	Incrementing Pattern (Hexadecimal)	
00	00 00 00 00	
01	00 00 01 01	
02	00 00 02 02	
03	00 00 03 03	
04	00 00 00 04	Bits 16 through 21
05	00 00 01 05	must always
|	|	be clear
FE	00 00 02 FE	
FF	00 00 03 FF	

C.1 a. Sweep locations 00 through FF of BDP PAK 0 EBCDIC RAM.
b. Read, format and report BDP Parity Status; error if not 0--0.

Parity Status Format:

See section 18 Description.

Section 18, Subsection 2

This subsection tests BDP RAM PAK 0 EBCDIC memory addressing logic.

Method:

The BDP EBCDIC RAMs are built from 1 by 256 bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

Conditions:

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in BDP PAK 0 EBCDIC RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in BDP PAK 0 EBCDIC RAM, location X+1 if X is not FF(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00 Sweep locations 0 through X-1.
b. Read, format and report BDP Parity Status; error if not 0--0.

C.4
a. If Test Address X is not FF(hex), Sweep locations X+1 through FF(hex).
b. Read, format and report BDP Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report BDP Parity Status; error if BDP Parity Status is not equal Expected Parity Status.

- Write Test Address X location with Base pattern.
- If Repeat Iteration set, go to C.1.
- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.
- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, clean up memory:
  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear errors in processor.

Test Patterns (Hexadecimal):

No.	Test Address X	Test Data Pattern	Parity Status
0	00	7F 00 00 00	00
1	00	00 7F 00 00	00
2	00	00 00 01 00	04
3	00	00 00 00 7F	04
4	00	80 00 00 00	00
5	00	00 80 00 00	00
6	00	00 00 02 00	04
7	00	00 00 00 80	04
8	55	00 00 01 7F	04
9	AA	00 00 01 7F	04
A	FF	00 00 01 7F	04

Base Patterns (Hexadecimal):

0	00 00 02 80

Parity Status Format:

See Section 18 Description.

Section 18, Subsection 3

This subsection tests the addressing circuitry internal to the BDP RAM PAK 0 EBCDIC memory chips.

Method:

The BDP EBCDIC RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping BDP memories.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2
Report Test Address X in Condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in BDP PAK 0 EBCDIC RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

If Repeat Iteration set, repeat C.3.

C.4
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report so report 0--0.

If Test Address X is not equal to FF(hex), increment Test Address X and repeat C.2 through C.4.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	00 00 01 7F
01	|
02	|
incrementing address	Same data pattern
FE	|
FF	00 00 01 7F

Base Patterns (Hexadecimal):

0 00 00 02 80

Parity Status Format:

See Section 18 Description.

SECTION 19 - TEST BDP RAM PAK 0 CONVERT MEMORY

This section completely verifies the data holding and addressing capabilities of the BDP RAM PAK 0 CONVERT memory logic (memory B).

Subsection 0 - Tests BDP RAM PAK 0 Convert memory chips with various data patterns, all locations.

Subsection 1 - Tests BDP RAM PAK 0 Convert memory read data parity checkers.

Subsection 2 - Tests BDP RAM PAK 0 Convert memory addressing logic.

Subsection 3 - Tests the addressing circuitry internal to the BDP RAM PAK 0 Convert memory chips.

Hardware Tested:

- The BDP RAM PAK 0 Convert memory chips.
- The BDP RAM PAK 0 Convert memory chip read data parity checkers.
- The BDP RAM PAK 0 Convert memory addressing logic.

Method:

See the individual subsection Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At		PFS Register No.	PFS Register Bit
63	BDP RAM PAK 0 - B Data	(loc A3A2)	81	11
62	BDP RAM PAK 0 - B Address	(loc A3A2)	81	10
61	BDP RAM PAK 0 - A Data	(loc A3A2)	81	9
60	BDP RAM PAK 0 - A Address	(loc A3A2)	81	8
59	BDP Port B Stage 2 Rgtr	(loc A3C1)	81	5
58	BDP Port A Stage 2 Rgtr	(loc A3B1)	81	4
57	BDP RAM PAK 1 Data	(loc A3B2)	81	3
56	BDP RAM PAK 1 Address	(loc A3B2)	81	2
55	BDP Port A RAM Address	(loc A3B1)	80	55
54	BDP Port B RAM Address	(loc A3C1)	80	54

Section 19, Subsection 0

This subsection tests BDP RAM PAK 0 Convert memory chips with various data patterns, all locations.

Method:

All locations in the BDP PAK 0 Convert RAM memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the BDP PAK 0 Convert RAM locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of BDP PAK 0 Convert RAM; locations 00 through FF(hex).
b. Read, format and report BDP Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	00 03 FF FF
2	00 02 AA AA
3	00 01 55 55
4	00 03 7F 7F
5	00 00 80 80
6	00 00 00 00

Parity Status Format:

See Section 19 Description.

Section 19, Subsection 1

This subsection tests BPD RAM PAK 0 Convert memory read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the BDP PAK 0 Convert RAM read data parity checkers. This pattern is written into the RAM memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc). The RAM memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it shouldn't.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0 Write all locations of BDP PAK 0 Convert RAM with an incrementing pattern, beginning at location 0 and ending with location FF(hex); no valid data to report so report 0--0.

Location	Incrementing Pattern (Hexadecimal)	
00	00 00 00 00	
01	00 01 01 01	
02	00 02 02 02	
03	00 03 03 03	
04	00 00 04 04	Bits 8 through 13
05	00 01 05 05	must always
|	| |	be clear
FE	00 02 FE FE	
FF	00 03 FF FF	

C.1 a. Sweep locations 00 through FF of BDP PAK 0 Convert RAM.
b. Read, format and report BDP Parity Status; error if not 0--0.

Parity Status Format:

See section 19 Description.

Section 19, Subsection 2

This subsection tests BDP RAM PAK 0 Convert memory addressing logic.

Method:

The BDP Convert RAMs are built from 1 by 256 bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

Conditions:

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in BDP PAK 0 Convert RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in BDP PAK 0 Convert RAM, location X+1 if X is not FF(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear Errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00 Sweep locations 0 through X-1.
b. Read, format and report BDP Parity Status; error if not 0--0.

C.4
a. If Test Address X is not FF(hex), Sweep locations X+1 through FF(hex).
b. Read, format and report BDP Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report BDP Parity Status; error if BDP Parity Status is not equal Expected Parity Status.

- Write Test Address X location with Base pattern.
- If Repeat Iteration set, go to C.1.
- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.
- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, clean up memory:
  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear errors in processor.

Test Patterns (Hexadecimal):

No.	Test Address X	Test Data Pattern	Expected Parity Status
0	00	7F 00 00 00	00
1	00	00 01 00 00	01
2	00	00 00 7F 00	01
3	00	00 00 00 7F	01
4	00	80 00 00 00	00
5	00	00 02 00 00	01
6	00	00 00 80 00	01
7	00	00 00 00 80	01
8	55	00 01 7F 7F	01
9	AA	00 01 7F 7F	01
A	FF	00 01 7F 7F	01

Base Patterns (Hexadecimal):

0 00 02 80 80

Parity Status Format:

See Section 19 Description.

Section 19, Subsection 3

This subsection tests the addressing circuitry internal to the BDP RAM PAK 0 Convert memory chips.

Method:

The BDP Convert RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping BDP memories.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2
Report Test Address X in Condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in BDP PAK 0 Convert RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

If Repeat Iteration set, repeat C.3.

C.4
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report so report 0--0.

If Test Address X is not equal to FF(hex), increment Test Address X and repeat C.2 through C.4.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	00 01 7F 7F
01	|
02	|
incrementing address	Same data pattern
FE	|
FF	00 01 7F 7F

Base Patterns (Hexadecimal):

0 00 02 80 80

Parity Status Format:

See Section 19 Description.

SECTION 20 - TEST BDP RAM PAK 1 SPEC ERROR MEMORY

This section completely verifies the data holding and addressing capabilities of the BDP RAM PAK 1 SPEC ERROR memory logic (memory A).

Subsection 0 - Tests BDP RAM PAK 1 Spec Error memory chips with various data patterns, all locations.

Subsection 1 - Tests BDP RAM PAK 1 Spec Error memory read data parity checkers.

Subsection 2 - Tests BDP RAM PAK 1 Spec Error memory addressing logic.

Subsection 3 - Tests the addressing circuitry internal to the BDP RAM PAK 1 Spec Error memory chips.

Hardware Tested:

- The BDP RAM PAK 1 Spec Error memory chips.
- The BDP RAM PAK 1 Spec Error memory chip read data parity checkers.
- The BDP RAM PAK 1 Spec Error memory addressing logic.

Method:

See the individual subsection Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At		PFS Register No.	PFS Register Bit
63	BDP RAM PAK 0 - B Data	(loc A3A2)	81	11
62	BDP RAM PAK 0 - B Address	(loc A3A2)	81	10
61	BDP RAM PAK 0 - A Data	(loc A3A2)	81	9
60	BDP RAM PAK 0 - A Address	(loc A3A2)	81	8
59	BDP Port B Stage 2 Rgtr	(loc A3C1)	81	5
58	BDP Port A Stage 2 Rgtr	(loc A3B1)	81	4
57	BDP RAM PAK 1 Data	(loc A3B2)	81	3
56	BDP RAM PAK 1 Address	(loc A3B2)	81	2
55	BDP Port A RAM Address	(loc A3B1)	80	55
54	BDP Port B RAM Address	(loc A3C1)	80	54

Section 20, Subsection 0

This subsection tests BDP RAM PAK 1 Spec Error memory chips with various data patterns, all locations.

Method:

All locations in the BDP PAK 1 Spec Error RAM memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the BDP PAK 1 Spec Error RAM locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of BDP PAK 1 Spec Error RAM (locations 00 through FF(hex)).
b. Read, format and report BDP Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	00 00 03 FF
2	00 00 02 AA
3	00 00 01 55
4	00 00 03 7F
5	00 00 00 80
6	00 00 00 00

Parity Status Format

See Section 20 Description

Section 20, Subsection 1

This subsection tests BDP RAM PAK 1 Spec Error memory read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the BDP PAK 1 Spec Error RAM read data parity checkers. This pattern is written into the RAM memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc). The RAM memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it should not.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0 Write all locations of BDP PAK 1 Spec Error RAM with an incrementing pattern, beginning at location 0 and ending with location FF(hex); no valid data to report so report 0--0.

Location	Incrementing Pattern (Hexadecimal)	
00	00 00 00 00	
01	00 00 01 01	
02	00 00 02 02	
03	00 00 03 03	
04	00 00 00 04	Bits 16 through 21
05	00 00 01 05	must always
|	| |	be clear.
FE	00 00 02 FE	
FF	00 00 03 FF	

C.1 a. Sweep locations 00 through FF of BDP PAK 1 Spec Error RAM.
b. Read, format and report BDP Parity Status; error if not 0--0.

Parity Status Format

See section 20 Description.

Section 20, Subsection 2

This subsection tests BDP RAM PAK 1 Spec Error memory addressing logic.

Method:

The BDP Spec Error RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

Method:

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

Conditions:

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in BDP PAK 1 Spec Error RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in BDP PAK 1 Spec Error RAM, location X+1 if X is not FF(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear Errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00 Sweep locations 0 through X-1.
b. Read, format and report BDP Parity Status; error if not 0--0.

C.4
a. If Test Address X is not FF(hex), Sweep locations X+1 through FF(hex).
b. Read, format and report BDP Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report BDP Parity Status; error if BDP Parity Status is not equal Expected Parity Status.

- Write Test Address X location with Base pattern.

- If Repeat Iteration set, go to C.1.

- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.

- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, without error, clean up memory:

  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear Errors in processor.

Test Patterns (Hexadecimal):

Number	Test Address X	Test Data Pattern	Parity Status
0	00	7F 00 00 00	00
1	00	00 7F 00 00	00
2	00	00 00 01 00	40
3	00	00 00 00 7F	40
4	00	80 00 00 00	00
5	00	00 80 00 00	00
6	00	00 00 02 00	40
7	00	00 00 00 80	40
8	55	00 00 01 7F	40
9	AA	00 00 01 7F	40
A	FF	00 00 01 7F	40

Base Patterns (Hexadecimal):

0 00 00 02 80

Parity Status Format

See Section 20 Description.

Section 20, Subsection 3

This subsection tests the addressing circuitry internal to the BDP RAM PAK 1 Spec Error memory chips.

Method:

The BDP Spec Error RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping BDP memories.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0 a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1 Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2 Report Test Address X in Condition Result byte 7; informational.

C.3 a. Sweep Test Address X location in BDP PAK 1 Spec Error RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

If Repeat Iteration set, repeat C.3.

C.4 a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report so report 0--0.

If Test Address X is not equal to FF(hex), increment Test Address X and repeat C.2 through C.4.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	00 00 01 7F
01	|
02	|
incrementing address	Same data pattern
FE	|
FF	00 00 01 7F

Base Patterns (Hexadecimal):

0 00 00 02 80

Parity Status Format:

See Section 20 Description.

SECTION 21 - TEST BDP RAM PAK 1 X256 MEMORY

This section completely verifies the data holding and addressing capabilities of the BDP RAM PAK 1 X256 memory logic (memory A).

Subsection 0 - Tests BDP RAM PAK 1 X256 memory chips with various data patterns, all locations.

Subsection 1 - Tests BDP RAM PAK 1 X256 memory read data parity checkers.

Subsection 2 - Tests BDP RAM PAK 1 X256 memory addressing logic.

Subsection 3 - Tests the addressing circuitry internal to the BDP RAM PAK 1 X256 memory chips.

Hardware Tested:

- The BDP RAM PAK 1 X256 memory chips.
- The BDP RAM PAK 1 X256 memory chip read data parity checkers.
- The BDP RAM PAK 1 X256 memory addressing logic.

Method:

See the individual subsection Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At		PFS Register No.	Bit
63	BDP RAM PAK 0 - B Data	(loc A3A2)	81	11
62	BDP RAM PAK 0 - B Address	(loc A3A2)	81	10
61	BDP RAM PAK 0 - A Data	(loc A3A2)	81	9
60	BDP RAM PAK 0 - A Address	(loc A3A2)	81	8
59	BDP Port B Stage 2 Rgtr	(loc A3C1)	81	5
58	BDP Port A Stage 2 Rgtr	(loc A3B1)	81	4
57	BDP RAM PAK 1 Data	(loc A3B2)	81	3
56	BDP RAM PAK 1 Address	(loc A3B2)	81	2
55	BDP Port A RAM Address	(loc A3B1)	80	55
54	BDP Port B RAM Address	(loc A3C1)	80	54

Section 21, Subsection 0

This subsection tests BDP RAM PAK 1 X256 memory chips with various data patterns, all locations.

Method:

All locations in the BDP PAK 1 X256 RAM memory chips are written with a test pattern. The memory chips are then swept with a sweep function (all locations read out in sequence). Since the memory data cannot be read by the MCH, the only indication of a bit failure is a parity error. The memory's parity status is then read from PFS Registers and verified that no data parity errors occurred during the sweep.

Different data patterns are used in order to thoroughly test the memory chips' capability to hold data in all locations.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write all of the BDP PAK 1 X256 RAM locations with Test Pattern.
b. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

C.1
a. Sweep all locations of BDP PAK 1 X256 RAM (locations 00 through FF(hex)).
b. Read, format and report BDP Parity Status; error if not 0--0.

Exit subsection if end of Test Patterns.

Test Patterns (Hexadecimal):

0	00 00 00 00
1	00 03 FF FF
2	00 02 AA AA
3	00 01 55 55
4	00 03 7F 7F
5	00 00 80 80
6	00 00 00 00

Parity Status Format:

See Section 21 Description

Section 21, Subsection 1

This subsection tests BPD RAM PAK 1 X256 memory read data parity checkers.

Method:

An incrementing data pattern is used to verify the correct operation of the BDP PAK 1 X256 RAM read data parity checkers. This pattern is written into the RAM memory chips in successive locations (location 0 contains 00, location 1 contains 01, etc). The RAM memory is then swept. If the parity checkers do not function correctly, at least one of the incrementing pattern data words will cause the faulty parity checker to report a parity error when it shouldn't.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0 Write all locations of BDP PAK 1 X256 RAM with an incrementing pattern, beginning at location 0 and ending with location FF(hex); no valid data to report so report 0--0.

Location	Incrementing Pattern (Hexadecimal)	
00	00 00 00 00	
01	00 01 01 01	
02	00 02 02 02	
03	00 03 03 03	
04	00 00 04 04	Bits 8 through 13
05	00 01 05 05	must always
|	| |	be clear.
FE	00 02 FE FE	
FF	00 03 FF FF	

C.1 a. Sweep locations 00 through FF of BDP PAK 1 X256 RAM.
b. Read, format and report BDP Parity Status; error if not 0--0.

Parity Status Format:

See section 21 Description.

Section 21, Subsection 2

This subsection tests BDP RAM PAK 1 X256 memory addressing logic.

Method:

The BDP X256 RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a modified March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty, and using a parity error to indicate when the correct address has been read.

Method:

The basic test procedure, using a modified March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Write one location (Test Address X) with a special test pattern containing bad parity.

c. Sweep all locations in the memory, except the one with bad parity, and verify that no data parity error occurred.

d. Sweep the location containing bad parity and verify that a parity error does occur.

Write the bad parity data location with good base pattern data and then repeat steps b through d using next Test Address to hold bad parity data. Steps b through d will be repeated until all test locations have contained bad parity data one time.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.
- Write all locations in memory with Base Pattern 0.

Conditions:

C.0 Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

Conditions:

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X in BDP PAK 1 X256 RAM (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Report Test Address X; informational.

C.2
a. Sweep one good location in BDP PAK 1 X256 RAM, location X+1 if X is not FF(hex), else location 00. This clears out bad parity data from input to the parity checkers.
b. Clear Errors in processor.
c. Report Test Data Pattern number in Condition Result byte 1 and Test Data Pattern data in Condition Result bytes 4 through 7; informational.

C.3
a. If Test Address X not 00 Sweep locations 0 through X-1.
b. Read, format and report BDP Parity Status; error if not 0--0.

C.4
a. If Test Address X is not FF(hex), Sweep locations X+1 through FF(hex).
b. Read, format and report BDP Parity Status; error if not 0--0.

C.5 Report Expected Parity Status for Test Data Pattern; informational.

C.6
a. Sweep Test Address X location.
b. Read, format and report BDP Parity Status; error if BDP Parity Status is not equal Expected Parity Status.

- Write Test Address X location with Base pattern.

- If Repeat Iteration set, go to C.1.

- If more Test Addresses/Test Data Patterns, update Test Address and Test Data Patterns and repeat C.0 through C.6.

- When all Test Addresses/Test Data Patterns and Base Patterns have been executed, clean up memory:

  - Write all memory locations with 0--0 data.
  - Sweep location 0 of memory.
  - Clear errors in processor.

Test Patterns (Hexadecimal):

Number	Test Address X	Test Data Pattern	Expected Parity Status
0	00	7F 00 00 00	00
1	00	00 01 00 00	40
2	00	00 00 7F 00	40
3	00	00 00 00 7F	40
4	00	80 00 00 00	00
5	00	00 02 00 00	40
6	00	00 00 80 00	40
7	00	00 00 00 80	40
8	55	00 01 7F 7F	40
9	AA	00 01 7F 7F	40
A	FF	00 01 7F 7F	40

Base Patterns (Hexadecimal):

0 00 02 80 80

Parity Status Format:

See Section 21 Description.

Section 21, Subsection 3

This subsection tests the addressing circuitry internal to the BDP RAM PAK 1 X256 memory chips.

Method:

The BDP X256 RAMs are built from 1 by 256-bit memory chips, one chip for each bit position in the memory data. Since the data cannot be read and inspected by the MCH, a sweep operation must be used and the parity checkers relied on to detect any addressing errors. This requires the use of a forward March Pattern testing technique, using special data patterns that will cause parity errors to occur when addressing logic is faulty.

The basic test procedure, using a March Pattern, is as follows:

a. Write the entire memory with a base pattern (all locations contain same data).

b. Sweep one location (Test Address X) and verify that no data parity error occurred. Note that the hardware also reads data out of the subsequent location (X+1) when sweeping BDP memories.

c. Write location just swept with special test pattern containing bad parity.

Steps b and c are repeated using incrementing test addresses until a parity error is sensed, or all locations have been swept and written.

In order to thoroughly test some memories, more than one base pattern is sometimes required, as well as different bad parity test patterns.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0
a. Write all locations in the memory with Base Pattern.
b. Report Base Pattern number in Condition Result byte 1 and Base Pattern data in Condition Result bytes 4 through 7; informational.

C.1
Report Test Pattern data in Condition Result bytes 4 through 7; informational.

C.2
Report Test Address X in Condition Result byte 7; informational.

C.3
a. Sweep Test Address X location in BDP PAK 1 X256 RAM.
b. Clear residual parity errors in processor.
c. Sweep Test Address X location second time.
d. Read, format and report Parity Status; error if not 0--0.

If Repeat Iteration set, repeat C.3.

C.4
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Test Data Pattern to Test Address X (Test Data Pattern will be written with bad parity).
c. Clear bit 18 of PTM Register.
d. Nothing valid to report so report 0--0.

If Test Address X is not equal to FF(hex), increment Test Address X and repeat C.2 through C.4.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	00 01 7F 7F
01	|
02	|
incrementing address	Same data pattern
FE	|
FF	00 01 7F 7F

Base Patterns (Hexadecimal):

0 00 02 80 80

Parity Status Format:

See Section 21 Description.

SECTION 22 - TEST REGISTER FILE

This section verifies the error reporting, data holding and addressing capabilities of the Register File Auxiliary boards and the MAC Register File read path.

Subsection 0 - Tests parity error reporting from the Register File read data parity checkers and the MAC Register File byte selection logic.

Subsection 1 - Tests the Register File with various data patterns, all locations, and the Control Store Disassembly Network byte selection logic.

Subsection 2 - Tests the Register File read data parity checkers and addressing logic external to the Register File memory chips.

Subsection 3 - Tests the addressing circuitry both external and internal to the Register File memory chips.

Hardware Tested:

- The MAC Register File Byte Select Decoder, Register File Go logic and other logic related to MAC reading and writing the Register File on the MAC BZ and BK ZIF boards.
- The MAC byte select logic on the Register File Auxiliary boards (8DDH auxiliary boards at B3A1 and B3A2).
- The Register File Read Data Register parity checkers.
- The corresponding parity status bits in the PFS Register (PFS Register 84 bits 57 and 58).
- The Register File memory chips on 8DDH auxiliary boards at B3A1 and B3A2.
- The Control Store Disassembly Network (Register File data path), all logic contained on LSI Arrays.
- The Register File memory addressing logic.

Method:

See the individual subsection Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	Register File Data bits 32 through 63/ 68 through 71 (loc B3A2)	84	58
62	Register File Data bits 0 through 31/ 64 through 67 (loc B3A1)	84	57
61	MAC Data Mux 5 (loc C1D)	86	35

Section 22, Subsection 0

This subsection tests parity error reporting from the Register File read data parity checkers and the MAC Register File byte selection logic.

Method:

Location 0 of the Register File is written with 64 bits of data (8 bytes), one byte of which contains bad parity. Location 0 is then read and the parity status bits read from PFS Register 84 to verify that a parity error has been reported by the Register File read data parity checkers. This sequence is repeated with each byte containing bad parity once to ensure that each of the 8 parity checkers will report an error.

The Register File byte selection logic must operate properly, or one of the test patterns will not cause a parity error to be reported when it should (the byte with bad parity will have been overwritten with good byte data), or a parity error will be reported but the read data will indicate different data than was written.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write Register File location 0 with 8 byte Test Pattern (one byte will normally contain bad parity).
c. Clear bit 18 of PTM Register.
d. Write Register File location 1 with 8 bytes of 00 (good parity).
e. Clear errors in processor.
f. Report Test Pattern number in Condition Result byte 7; informational.

Conditions:

C.1 Report Test Pattern data; informational. Byte with odd number of bits set is byte that will have bad parity.

C.2 Read Register File location 0, 8 bytes, and report data read; error if data read not equal Test Pattern data.

C.3 Report Expected Parity Status for Test Pattern used; informational.

C.4 Read, format and report Register File parity status; error if not equal Expected Parity Status.

Exit subsection if no more test patterns.

Test Patterns (Hexadecimal):

Number	Test Patterns	Expected Parity Status
0	01FF FFFF FFFF FFFF	06
1	FF01 FFFF FFFF FFFF	06
2	FFFF 01FF FFFF FFFF	06
3	FFFF FF01 FFFF FFFF	06
4	FFFF FFFF 01FF FFFF	05
5	FFFF FFFF FF01 FFFF	05
6	FFFF FFFF FFFF 01FF	05
7	FFFF FFFF FFFF FF01	05
8	0000 0000 0000 0000	00

Parity Status Format:

See Section 22 Description.

Section 22, Subsection 1

This subsection tests the Register File with various patterns, all locations, and the Control Store Disassembly Network byte selection logic.

Method:

All locations of the Register File are written with a test pattern, same data pattern in every location. Each location is then read to verify that is contains the proper data. The parity status is also read to verify that no parity errors occur. Numerous data patterns are used to verify the data holding capabilities of the Register File memory chips.

Note that whenever a Register File location is read, the subsequent location's data is also parity checked. Thus, the location read may contain good data but a parity error is reported because the next location contains bad data. Because of this, C.5 will report and check data from the subsequent Register File location whenever a parity error is reported by C.4.

Two special byte incrementing patterns are included to verify the proper operation of byte selection logic at the Control Store Disassembly Network.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write every location in the Register File (00 through 3F(hex)) with the 8 byte Test Pattern.
b. Report the Test Pattern number in Condition Result byte 7; informational.

C.1
a. Initialize Register File Location Pointer to 00.
b. Report the 8 byte Test Pattern data; informational.

C.2
a. Read Register File location per Location Pointer.
b. Report data read; error if not equal Test Pattern data written in C.0.

C.3 Report the value of the Location Pointer; informational

C.4 Read, format and report Register File parity status; error if not 0.

If C.2 and C.4 did not fail and Location Pointer value is not equal 3F(hex), increment Location Pointer value and repeat C.2 through C.4.

If C.4 failed, execute C.5; else skip C.5.

C.5
a. Read Register File location per Location Pointer +1.
b. Report data read; error if not equal Test Pattern data written in C.0.

If Repeat Iteration set, go to C.2.

Exit subsection if no more test patterns.

Test Patterns (Hexadecimal):

0	0000 0000 0000 0000	
1	FFFF FFFF FFFF FFFF	
2	AAAA AAAA AAAA AAAA	
3	5555 5555 5555 5555	
4	7F7F 7F7F 7F7F 7F7F	
5	8080 8080 8080 8080	Tests even/odd byte parity
6	7F00 7F00 7F00 7F00	bit placement.
7	007F 007F 007F 007F	
8	0011 2233 4455 6677	Tests byte select control
9	FFEE DDCC BBAA 9988	of CS Disassembly Network.
A	0000 0000 FFFF FFFF	
B	FFFF FFFF 0000 0000	Sensitizes CS
C	00FF 00FF 00FF 00FF	Disassembly Network
D	FF00 FF00 FF00 FF00	

Parity Status Format:

See Section 22 Description.

Section 22, Subsection 2

This subsection tests the Register File read data parity checkers and addressing logic external to the Register File memory chips.

Method:

An incrementing pattern is written into each consecutive location of the Register File: e.g.; location 0 equals 0000 0000 0000 000(hex), location 1 equals 0101 0101 0101 0101(hex), etc. The entire Register File is then read, one location at a time, and the data from each location is compared to what was written into it. Any data miscompare, which will indicate some type of addressing problem, will be reported as an error.

After each location of the Register File has been read, the Register File parity status bits are read from the PFS Register (PFS Register 84 bits 57 and 58) and verified that none of the Register File read data parity checkers reported a parity error.

Note that whenever a Register File location is read, the subsequent location's data is also parity checked. Thus, the location read may contain good data but a parity error is reported because the next location contains bad data. Because of this, C.5 will report and check data from the subsequent Register File location whenever a parity error is reported by C.4.

The above two steps are repeated with three other sets of incrementing patterns in order to test all eight parity checkers with every combination of eight bit data bytes (00, 01, ..., FE, FF).

Initialization:

- Send Master Clear.

Repeat the following conditions for each set of Test Patterns.

Conditions:

C.0
a. Write each location of the Register File with an incrementing data pattern per the Test Pattern Set, location 00(hex) through location 3F(hex).
b. Initialize Register File Location Pointer to 00.
c. Report the Test Pattern Set number in byte 7 of the Condition Result; informational.

C.1 Report Register File Location Pointer value; informational.

C.2 Report expected contents of the Register File, location per Location Pointer; informational.

C.3
a. Read 8 bytes of data from the Register File, location per Location Pointer.
b. Report data read from Register File; error if not equal expected contents reported in C.2.

C.4 Read, format and report Register File parity status; error if not equal 0.

If C.3 and C.4 did not fail and Location Pointer value is not equal 3F(hex), increment Location Pointer value and repeat C.1 through C.4.

If C.4 failed, execute C.5; else skip C.5.

C.5
a. Read Register File location per Location Pointer +1.
b. Report data read; error if not equal to data written into this location by C.0.

Exit subsection if error or no more test pattern sets.

Test Pattern Sets (Hexadecimal):

Set Number	Test Pattern	Written in Rgtr File Location
0	0000 0000 0000 0000	00
	0101 0101 0101 0101	01
	|	|
	incrementing pattern	|
	|	|
	3E3E 3E3E 3E3E 3E3E	3E
	3F3F 3F3F 3F3F 3F3F	3F
1	4040 4040 4040 4040	00
	4141 4141 4141 4141	01
	|	|
	incrementing pattern	|
	|	|
	7E7E 7E7E 7E7E 7E7E	3E
	7F7F 7F7F 7F7F 7F7F	3F
2	8080 8080 8080 8080	00
	8181 8181 8181 8181	01
	|	|
	incrementing pattern	|
	|	|
	BEBE BEBE BEBE BEBE	3E
	BFBF BFBF BFBF BFBF	3F
3	C0C0 C0C0 C0C0 C0C0	00
	C1C1 C1C1 C1C1 C1C1	01
	|	|
	incrementing pattern	|
	|	|
	FEFE FEFE FEFE FEFE	3E
	FFFF FFFF FFFF FFFF	3F

Parity Status Format:

See Section 22 Description.

Section 22, Subsection 3

This subsection tests the addressing circuitry both external and internal to the Register File memory chips.

Method:

The Register File consists of 4 by 16-bit memory chips, one chip for each four bit positions in the memory data. In order to thoroughly test the internal addressing circuitry in each of these 4 by 16-bit memory chips, a forward and reverse March Pattern testing technique with special data patterns is used. This technique also completes the testing of addressing logic external to the memory chips, but still on the Register File Auxiliary boards.

The basic test procedure, using a forward and reverse March Pattern, is as follows:

a. Write the entire Register File with a base pattern (all locations contain same data).
b. Read one location (Test Address X) and verify that the data has not been altered (equal base pattern) and that no parity error has occurred.
c. Write the location just read with a special test data pattern.

Step b and c are repeated using incrementing test addresses (00 to 3F(hex)) until a data miscompare and/or parity error is sensed, or all locations have been read and written in a forward direction.

d. Write the entire Register File with the base pattern.
e. Read one location (Test Address X) and verify that the data has not been altered (equal base pattern) and that no parity error occurred.
f. Write the location just read with the special test data pattern.

Steps e and f are repeated using decrementing test addresses (3F(hex) to 00) until a data miscompare and/or parity error is sensed, or all locations have been read and written in a reverse direction.

To thoroughly test the 4 by 16-bit memory chips, steps a through f are then repeated using a different base pattern.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0
a. Write all locations in the Register File with Base Pattern.
b. Report Base Pattern number in Condition Result byte 7; informational.

C.1
a. Initialize Test Address X to 00.
b. Report Base Pattern data; informational.

C.2 Report Test Pattern data; informational.

Conditions:

C.3 Report Test Address X value in Condition Result byte 7; informational.

C.4
a. Read Test Address X location from Register File.
b. Report data read from Test Address X; error if data is not equal Base Pattern.

C.5 Read, format and report Register File Parity Status; error if not 0.

C.6 If C.5 failed, Read and report data from Test Address X+1; error if data is not equal Base Pattern for addresses 01 through 3F(hex), Test Data Pattern for address 00.

If C.5 did not fail, nothing valid to report, so report 0--0.

C.7 If neither C.4 nor C.5 failed, write Test Data Pattern to Test Address X. Nothing valid to report, so report 0--0.

If neither C.4 nor C.5 failed and Test Address X is not equal to 3F(hex), increment Test Address X and repeat C.3 through C.6.

C.8
a. Write all locations in the Register File with Base Pattern.
b. Initialize Test Address X to 3F(hex).
c. Nothing valid to report, so report 0--0.

C.9 Report Test Address X value in Condition Result byte 7; informational.

C.10
a. Read Test Address X location from Register File.
b. Report data read from Test Address X; error if data is not equal Base Pattern.

C.11 Read, format and report Register File Parity Status; error if not 0.

C.12 If C.11 failed, Read and report data from Test Address X+1; error if data is not equal Test Data Pattern for addresses 01 through 3F(hex), Base Pattern for address 00.

If C.11 did not fail, nothing valid to report, so report 0--0.

C.13 If neither C.10 nor C.11 failed, write Test Data Pattern to Test Address X. Nothing valid to report, so report 0--0 .

If neither C.10 nor C.11 failed and Test Address X is not equal to 00, decrement Test Address X and repeat C.9 through C.13.

Exit subsection if no more base patterns.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
00	EEEE EEEE EEEE EEEE
01	
incrementing address	Same Data Pattern
3E	
3F	EEEE EEEE EEEE EEEE

Base Patterns (Hexadecimal):

0	1010 1010 1010 1010
1	0101 0101 0101 0101

Parity Status Format:

See Section 22 Description.

SECTION 23 - TEST CONTROL STORE BREAKPOINT AND MICRAND ADDRESS REGISTERS

This section verifies the data holding capabilities of both the Control Store Breakpoint Register and the Control Store Micrand Address Register, the MAC read and write paths to these Registers, the data error reporting and parity generation logic associated with these Registers.

Subsection 0 - Tests MAC capability to write and read the Control Store Breakpoint Register with various data patterns.

Subsection 1 - Tests the parity generator on the MAR/BKPT data path to MAC MUX2.

Subsection 2 - Tests MAC capability to write and read the Control Store Micrand Address Register with various data patterns, and test the CS Micrand Address Parity Regeneration logic.

Subsection 3 - Tests parity error reporting from the Control Store Micrand Address Register parity checker.

Subsection 4 - Tests the Control Store Micrand Address Register parity checker.

Hardware Tested:

- The MAC logic required to write and read the Control Store Breakpoint and Address Registers, such logic contained on both MAC ZIF boards (BZ at C1E and BK at C1D).
- The Control Store Breakpoint Register, Micrand Address Register, BKP/Address Output Disassembly Mux, Micrand Address Register Parity Checker, and MAR/BKPT Parity Generator; all contained on the Control Store 7BNH ZIF board at C1C.
- The corresponding parity status bits in the PFS Registers (PFS Register 86 bit 3).

Method:

See the individual subsection Method documentation.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	MAC Channel Input Data (loc C1D)	86	34
62	MAC Channel Output Data (loc C1E)	86	32
61	CS Mic Address Register (loc C1C)	86	3

Initialization:

- Write Control Store Micrand Address Register with F--F(hex). This sensitizes the Bkp/Address Output Disassembly Mux for testing the Control Store Breakpoint Register (Section 23, Subsection 0).

Section 23, Subsection 0

This subsection tests MAC capability to write and read the Control Store Breakpoint Register with various data patterns.

Method:

The Control Store Breakpoint Register is written with a data pattern. The contents of the Breakpoint Register are then read and compared to what was written. In addition, the Control Store Micrand Address and MAC parity status is read from PFS Register 86 and verified that no parity error occurred. Various data patterns are used in order to thoroughly test the write/read control logic and data paths/Registers.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0 a. Write Control Store Breakpoint Register with 8 byte Test Pattern.
b. Report Test Pattern number in byte 7 of Condition Result; informational.

C.1 Report Test Pattern data written to Breakpoint Register in C.0; informational.

C.2 Report Expected Read Data Pattern; informational.

C.3 Read and report the contents of the Control Store Breakpoint Register; error if not equal Expected Read Data Pattern.

Conditions:

C.4 Read, format and report CS Microand Address and MAC parity status; error if not 0.

Exit subsection if no more test patterns.

Test Patterns (Hexadecimal):

No.	Test Patterns	Expected Read Data Patterns
0	0000 0000 0000 0000	0000 0000 0000 0000
1	FFFF FFFF FFFF FFFF	0000 0000 0000 07FF
2	5555 5555 5555 5555	0000 0000 0000 0555
3	AAAA AAAA AAAA AAAA	0000 0000 0000 02AA
4	0000 0000 0000 07FF	0000 0000 0000 07FF
5	FFFF FFFF FFFF F800	0000 0000 0000 0000
6	0000 0000 0000 00FF	0000 0000 0000 00FF
7	0000 0000 0000 0700	0000 0000 0000 0700
8	0000 0000 0000 0000	0000 0000 0000 0000

Parity Status Format:

See Section 23 Description.

Section 23, Subsection 1

This subsection tests the parity generator on the MAR/BKPT data path to MAC MUX2.

Method:

The Control Store Breakpoint Register is written and read with data patterns that will exercise every gate within the CS MAR/BKPT read data parity generator. The parity status of the MAC Channel Output Data parity checker is checked after reading each data pattern word to verify that no parity error has occurred. The occurrence of a parity error indicates that the CS parity generator (BN ZIF board at C1C) is faulty, since the MAC Channel Output Data parity checker has previously tested good.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0 a. Write Control Store Breakpoint Register with 8 byte Test Pattern.
b. Report Test Pattern number in byte 7 of Condition Result; informational.

C.1 Report Test Pattern data written to Breakpoint Register in C.0; informational.

C.2 Read and report the contents of the Control Store Breakpoint Register; error if not equal Test Pattern data written in C.0.

C.3 Read, format and report CS Microand Address and MAC parity status; error if not 0.

Exit subsection if error or no more test patterns.

Test Patterns (Hexadecimal):

No.	Test Patterns
0	0000 0000 0000 0000
1	0000 0000 0000 0001
2	0000 0000 0000 0002
|	|
|	incrementing pattern
|	|
FE	0000 0000 0000 00FE
FF	0000 0000 0000 00FF

Parity Status Format:

See Section 23 Description.

Section 23, Subsection 2

This subsection tests the MAC capability to write and read the Control Store Microand Address Register with various data patterns, and test the CS Microand Address Parity Regeneration logic.

Method:

The Control Store Microand Address Register is written with a data pattern. The contents of the Microand Address Register are then read and compared to what was written. In addition, the Control Store Microand Address and MAC parity status is read from PFS Register 86 and verified that no parity error occurred. Various data patterns are used to thoroughly test the Control Store Microand Address Register and the Parity Regeneration logic.

Initialization:

- Send Master Clear and Clear Error Functions to processor.
- Write CS Breakpoint Register with F--F(hex) to sensitize MAR/BKPT Output Disassembly Mux.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write Control Store Microand Address Register with 8 byte Test Pattern.
b. Report Test Pattern number in byte 7 of Condition Result; informational.

C.1 Report Test Pattern data written to Microand Address Register in C.0; informational.

C.2 Read and report the contents of the Control Store Microand Address Register; error if not equal Test Pattern data.

C.3 Read, format and report CS Microand Address and MAC parity status; error if not 0.

Exit subsection if no more test patterns.

Test Patterns (Hexadecimal):

0	0000 0000 0000 0000
1	0000 0000 0000 07FF
2	0000 0000 0000 0555
3	0000 0000 0000 02AA
4	0000 0000 0000 0780
5	0000 0000 0000 0580
6	0000 0000 0000 00FF
7	0000 0000 0000 0700
8	0000 0000 0000 0000

Parity Status Format:

See Section 23 Description.

Section 23, Subsection 3

This subsection tests parity error reporting from the Control Store Microand Address Register parity checker.

Method:

The Control Store Microand Address Register is written with bad parity data. The corresponding PFS Register bit is then read and verified that it has been set (PFS Register 86, bit 3).

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Conditions:

C.0
a. Write Control Store Microand Address Register with bad parity data pattern (0000 0000 0000 1000(hex)). Bit 2**12 being set will cause a parity error in Microand Address.
b. Read, format and report CS Microand Address and MAC parity status; error if all bits are not clear except PFS 86 through bit 3.

C.1
a. Write Control Store Microand Address Register with 0--0.
b. Clear Errors in processor.
c. Read, format and report CS Microand Address and MAC parity status; error if not equal to 0.

Parity Status Format:

See Section 23 Description.

Section 23, Subsection 4

This subsection tests the Control Store Microand Address Register parity checker.

Method:

The Control Store Microand Address Register parity checker is a special 11/1 bit parity checker which is unique in processor. It is composed of one 2K (12160) chip and two =1 (12107) chips. In order to thoroughly test this parity checker, two sets of test patterns are employed, one set to fully test the 2K (12160) chip and the other set to test the combination of the two =1 (12107) chips and the 2K (12160) chip.

The basic test procedure is to write the Microrand Address Register with each pattern from the two sets of test patterns, checking the PFS Register after the writes have been completed to verify that no parity error has been falsely reported by the parity checker.

Initialization:

- Send Master Clear and Clear Error Functions to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0 a. Write Control Store Microrand Address Register with Test Pattern data.
b. Report the Test Pattern number in Condition Result bytes 6 and 7; informational.

C.1 Report Test Pattern Data written in C.0; informational.

C.2 Read, format and report CS Microrand Address and MAC parity status; error if not 0.

Exit subsection if no more test patterns.

Test Patterns (Hexadecimal):

Number	Test Pattern X
0	0000 0000 0000 0000
1	0000 0000 0000 0004
2	0000 0000 0000 0100
3	0000 0000 0000 0104
4	0000 0000 0000 0002
5	0000 0000 0000 0001
6	0000 0000 0000 0003
7	0000 0000 0000 0102

Number	Test Pattern Y (incrementing bits 53 through 60)
8	0000 0000 0000 0000
9	0000 0000 0000 0008
A	0000 0000 0000 0010
B	0000 0000 0000 0018
C	0000 0000 0000 0020
|	|
|	|
105	0000 0000 0000 07E8
106	0000 0000 0000 07F0
107	0000 0000 0000 07F8

Parity Status Format:

See Section 23 Description.

SECTION 24 - TEST CONTROL STORE MEMORY

This section verifies the error reporting, data holding and addressing capabilities of the Control Store memory ZIF boards; and the MAC Control Store write and read data paths. Each subsection will be executed 4 times. Each pass through a subsection will test the lower 2K and upper 2K of both the control store memory and the CS shadow memory.

Subsection 0 - Tests parity error reporting from the Control Store memory data parity checkers and the MAC Control Store byte selection logic.

Subsection 1 - Tests the Control Store memory with various data patterns, all locations; the Control Store Microrand Address Incrementer logic; and the MAC Control Store Read Data path.

Subsection 2 - Tests the Control Store memory data parity checkers and partial test of the addressing logic.

Subsection 3 - Tests the Control Store memory addressing logic both external and internal to the Control Store memory chips.

Hardware Tested:

- The MAC control signals which control the writing and reading of Control Store memory (MAC BZ and BK ZIF boards).
- The Control Store write byte enables and Select FU Microrand signal fanouts.
- The Control Store data parity checkers (LSI Arrays and BN ZIF board) and the corresponding PFS Register bits (PFS Register 86-bits 4 through 29).
- The Control Store memory chips on the BD ZIF boards at C1A and C1B (1x1024 bit memory chips).
- The Control Store FU Microrand Buffer Register, FU Microrand Register, and the Control Store FU data leg on the Control Store Disassembly Network, all composed of LSI Arrays.
- The Control Store Microrand Address Incrementer and its related logic on the BN ZIF board at C1C.

Method:

See the individual subsection Method documentation.

NOTE

The Processor Control Store RAM chips have poor write recovery. This sometimes causes the CS Read Data parity checkers to detect a parity error during a Control Store write operation. Because of this, Engineering has directed that the parity status bits from the CS Read Data parity checkers be ignored during CS write operations. CS Read Data parity status is only checked after CS read operations.

Condition Result Bit	Parity Error At		PFS Register No.	PFS Register Bit
63	MAC Data Mux 5	(loc C1D)	86	35
62	MAC Chan Input Data	(loc C1D)	86	34
61	MAC Chan Output Rgtr	(loc C1E)	86	32
60	CS Gen Micr byte 7	(loc C2H13)	86	29
59	CS Gen Micr byte 6	(loc C2H13)	86	28
58	CS Gen Micr byte 5	(loc C2H06)	86	27
57	CS Gen Micr byte 4	(loc C2H12)	86	26
56	CS Gen Micr byte 3	(loc C2H10)	86	25
55	CS Gen Micr byte 2	(loc C2H06)	86	24
54	FU Micr Register byte 7	(loc C2G14)	86	23
53	FU Micr Register byte 6	(loc C2G14)	86	22
52	FU Micr Register byte 5	(loc C2G05)	86	21
51	FU Micr Register byte 4	(loc C2G11)	86	20
50	FU Micr Register byte 3	(loc C2G11)	86	19
49	FU Micr Register byte 2	(loc C2G05)	86	18
48	FU Micr Register byte 1	(loc C2G02)	86	17
47	FU Micr Register byte 0	(loc C2G02)	86	16
46	FU Micr Buffer byte 7	(loc C2H14)	86	15
45	FU Micr Buffer byte 6	(loc C2H14)	86	14
44	FU Micr Buffer byte 5	(loc C2H05)	86	13
43	FU Micr Buffer byte 4	(loc C2H11)	86	12
42	FU Micr Buffer byte 3	(loc C2H11)	86	11
41	FU Micr Buffer byte 2	(loc C2H05)	86	10
40	FU Micr Buffer byte 1	(loc C2H02)	86	9
39	FU Micr Buffer byte 0	(loc C2H02)	86	8
38	MSC Register (8 through 15)	(loc C1C)	86	7
37	MSC Register (0 through 7)	(loc C1C)	86	6
36	MAC-CS odd RAM write data	(loc C1A)	86	5
35	MAC-CS even RAM write data	(loc C1B)	86	4
34	CS Microrand Address Reg	(loc C1C)	86	3

Initialization:

- Write all locations in the Register File with F--F(hex). Read Register File location 00 (no data check). This sensitizes the Control Store Output Disassembly Network.

Parity Status Format:

Same for all subsections.

Section 24, Subsection 0

This subsection tests parity error reporting from the Control Store memory data parity checkers and the MAC Control Store byte selection logic.

Method:

Location 0 of the Control Store memory is written with 128 bits of data (16 bytes: General Micrand and FU Micrand), one byte of which contains bad parity. Location 0 is then read. The Control Store parity status bits are read from PFS Register 86 to verify that a parity error has been reported from all appropriate parity checkers. This sequence is repeated with each byte containing bad parity once to ensure that all parity checkers in each byte position will report an error.

The Control Store write byte enables and associated control logic must operate correctly, or one of the test patterns will not cause a parity error to be reported when it should and/or data read will not be the same as written.

The Control Store read disassembly network has been previously tested for proper read byte selection in Section 22 (Register File test).

Initialization:

I.0 - Clear DEC bit 32.
I.1 - Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Clear errors in processor.
b. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
c. Write Control Store location 0 with 16-byte Test Pattern.
d. Clear bit 18 of PTM Register.
e. Report Test Pattern number; informational.

Conditions:

C.1 Report the General Micrand portion (8 bytes) of the Test Pattern written; informational.

C.2 Report the FU Micrand portion (8 bytes) of the Test Pattern written; informational.

C.3 Read, format and report Control Store parity status with MSC, General Micrand, FU Micrand and FU Micrand Buffer Register status bits masked off; error if not equal Expected Parity Status.

C.4
a. Clear processor errors.
b. Read 16 bytes of data from Control Store location 0.
c. Report General Micrand portion (8 bytes) of data read; error if not equal General Micrand portion of Test Pattern written.

C.5 Report FU Micrand portion (8 bytes) of data read in C.3; error if not equal FU Micrand portion of Test Pattern written.

C.6 Read, format and report Control Store parity status; error if status not equal Expected Parity Status.

If Repeat Iteration set, write zeros in Control Store Location 0 with good parity and repeat from condition 0.
If no more patterns for current portion of CS memory, increment flag.
If flag = CS upper 2K, set DEC bit 46 for upper 2K.
If flag = shadow , set DEC bit 32 for shadow memory.
go to I.1.

Exit subsection if no more test patterns.

Iteration Count	Num- ber	Test Pattern (Gen Mic)	Test Pattern (FU Mic)	Parity Status C.3	Parity Status C.6
0	12	0	FFFF FFFF FFFF FFFF	FFFF FFFF FFFF FFFF	0000 0003	0000 0000
1	13	1	01FF FFFF FFFF FFFF	FFFF FFFF FFFF FFFF	1800 0003	0501 0005
2	14	2	FF01 FFFF FFFF FFFF	FFFF FFFF FFFF FFFF	1800 0003	0280 8005
3	15	3	FFFF 01FF FFFF FFFF	FFFF FFFF FFFF FFFF	1800 0003	0040 4105
4	16	4	FFFF FF01 FFFF FFFF	FFFF FFFF FFFF FFFF	1800 0003	0020 2085
5	17	5	FFFF FFFF 01FF FFFF	FFFF FFFF FFFF FFFF	1880 0003	0010 1045
6	18	6	FFFF FFFF FF01 FFFF	FFFF FFFF FFFF FFFF	1800 0003	0008 0825
7	19	7	FFFF FFFF FFFF 01FF	FFFF FFFF FFFF FFFF	1800 0003	0004 0415
8	1A	8	FFFF FFFF FFFF FF01	FFFF FFFF FFFF FFFF	1800 0003	0002 020D
9	1B	9	FFFF FFFF FFFF FFFF	01FF FFFF FFFF FFFF	1800 0003	0501 0005
A	1C	A	FFFF FFFF FFFF FFFF	FF01 FFFF FFFF FFFF	1800 0003	0280 8005
B	1D	B	FFFF FFFF FFFF FFFF	FFFF 01FF FFFF FFFF	1800 0003	0040 4105
C	1E	C	FFFF FFFF FFFF FFFF	FFFF FF01 FFFF FFFF	1800 0003	0020 2085
D	1F	D	FFFF FFFF FFFF FFFF	FFFF FFFF 01FF FFFF	1800 0003	0010 1045
E	20	E	FFFF FFFF FFFF FFFF	FFFF FFFF FF01 FFFF	1800 0003	0008 0825
F	21	F	FFFF FFFF FFFF FFFF	FFFF FFFF FFFF 01FF	1800 0003	0004 0415
10	22	10	FFFF FFFF FFFF FFFF	FFFF FFFF FFFF FF01	1800 0003	0002 020D
11	23	11	0000 0000 0000 0000	0000 0000 0000 0000	0000 0003	0000 0000

NOTES

Iteration counts 12 - 23 tests shadow memory.

If no shadow memory is present (models 30, 31), iteration count will be 0 - 11.

Parity Status Format:

See section 24 Description.

Section 24, Subsection 1

This subsection tests the Control Store memory with various data patterns, all locations; the Control Store Microand Address Incrementer logic; and the MAC Control Store Read Data path.

Method:

The Control Store memory is written throughout with a test pattern, same data pattern in each location (128 bits per location). Each location is then read to verify that it contains the proper data. The parity status is also read to verify that no parity errors occurred during the write and read. Numerous data patterns are used to verify the data holding capabilities of the Control Store memory chips.

Note that whenever a Control Store location is read, the subsequent location's data is also parity checked. Thus, the location read may contain good data but a parity error is reported because the next location contains bad data. Because of this, C.10 and C.11 will report and check data from the subsequent Control Store location whenever a parity error is reported by C.9.

Two special byte incrementing patterns are included to verify the proper operation of the byte select logic for MAC writing and reading Control Memory.

A quick check of the Control Store Microand Address Incrementer logic is included during the writing of Control Store. The Control Store Microand Address Register is written with starting address 000. The Control Store is then fully written (locations 000 through 7FF(hex)) with the test pattern, using the Microand Address Incrementer to increment the Control Store Address for each successive 128-bit data word written. After the last Control

Store location has been written (7FF(hex)), the contents of the Microand Address Register are read. The contents should be 000, which verifies that the address has incremented from 000, 001, ..., 7FE(hex), 7FF(hex), 000; or never incremented at all. Section 24, Subsection 2 and 3 perform additional testing of the Microand Address Incrementer logic.

Initialization:

I.0 - Clear DEC bit 32.
I.1 - Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Write Control Store Microand Address Register with starting address of 000.
b. Write 800(hex) 128-bit words of Test Pattern to Control Store (fills locations 000 through 7FF(hex) of Control Stores memory and should cause Microand Address Register to end up containing 000).
c. Report Test Pattern number in Condition Result byte 7; informational.

C.1 Report General Microand portion (8 bytes) of Test Pattern written in Control Store; informational.

C.2 Report FU Microand portion (8 bytes) of Test Pattern written in Control Store; informational.

C.3 Read and report the contents of the Control Store Microand Address Register; error if not equal 000.

C.4
a. Initialize Control Store Location Pointer to 000.
b. Read, format and report Control Store parity status with MSC, General Microand, FU Microand and FU Microand Buffer Register status bits masked off; error if not 0.

C.5
a. Clear processor errors.
b. Write Microand Address Register with value of Control Store Location Pointer.
c. Nothing valid to report, so report 0--0.

C.6 Report value of Control Store Location Pointer; informational.

C.7
a. Read 128 bits (16 bytes) of data from Control Store using Control Word 2 equal to FF16 to sensitize CS Disassembly Mux, location per CS Microand Address Register (Address Register automatically incremented).
b. Report General Microand portion (8 bytes) of data read; error if not equal to General Microand portion of Test Pattern written in C.0.

C.8
a. Report FU Microand portion (8 bytes) of data read in C.7; error if not equal to FU Microand portion of Test Pattern written in C.0.

C.9 Read, format and report Control Store parity status; error if not 0.

If C.9 failed, execute C.10 and C.11, else skip them.

Conditions:

C.10 a. Read 128 bits (16 bytes) of data from Control Store location per CS Microand Address Register (contains subsequent address from address read by C.7).
b. Report General Microand portion (8 bytes) of data read; error if not equal General Microand portion of Test Pattern written in C.0.

C.11 a. Report FU Microand portion (8 bytes) of data read in C.10; error if not equal FU Microand portion of Test Pattern written in C.0.

If Repeat Iteration set, go to C.5.

Exit subsection if no more test patterns.

Otherwise, Increment Control Store Location Pointer and go to C.6.

If no more patterns for current portion of CS memory, increment flag.
If flag = CS upper 2K, set DEC bit 46 for upper 2K.
If flag = shadow , set DEC bit 32 for shadow memory.
Go to I.1.

Test Patterns (Hexadecimal):

No.	General Microand Portion	FU Microand Portion
0	0000 0000 0000 0000	0000 0000 0000 0000
1	FFFF FFFF FFFF FFFF	FFFF FFFF FFFF FFFF
2	5555 5555 5555 5555	5555 5555 5555 5555
3	AAAA AAAA AAAA AAAA	AAAA AAAA AAAA AAAA
4	8080 8080 8080 8080	8080 8080 8080 8080
5	0011 2233 4455 6677	8899 AABB CCDD EEFF
6	FFEE DDCC BBAA 9988	7766 5544 3322 1100

Parity Status Format:

See section 24 Description.

Section 24, Subsection 2

This subsection tests the Control Store memory data parity checkers and partially tests the addressing logic.

Method:

Locations 000 through 0FF(hex) of the Control Store memory are filled with an incrementing data pattern (location 000 contains 0--0, location 001 contains 0101--0101(hex), etc). Locations 000 through 0FF(hex) are then read, one location at a time, and the data read compared with what was written into the corresponding location. Any data miscompare will indicate some type of addressing problem and will be reported as an error.

After each location has been read, the Control Store parity status bits will be read and verified that none of the Control Store memory data parity checkers reported a parity error. A parity error report without bad read data indicates a faulty parity checker.

Note that whenever a Control Store location is read, the subsequent location's data is also parity checked. Thus, the location read may contain good data but a parity error is reported because the next location contains bad data. Because of this, C.8 and C.9 will report and check data from the subsequent Control Store location whenever a parity error is reported by C.6.

Initialization:

I.0 - Clear DEC bit 32.
I.1 - Send Master Clear and Clear Errors to processor.

Conditions:

C.0 a. Write locations 000 through 100(hex) with incrementing pattern.
b. Read, format and report Control Store parity status with MSC, General Microand, FU Microand and FU Microand Buffer Register status bits masked off; error if not zero.

C.1 a. Clear errors in processor.
b. Initialize Control Store Location Pointer with starting address of 000.
c. Write Control Store Microand Address Register with starting address of 000.
d. Nothing valid to report, so report 0--0.

C.2 Report Control Store Location Pointer value in Condition Result bytes 6 and 7; informational.

C.3 Report expected contents of Control Store, location of Control Store Location Pointer, same 64-bit data for both General and FU Microand portions of location; informational.

C.4 a. Read 128 bits (16 bytes) of data from Control Store, location per CS Microand Address Register (Address Register automatically increments).
b. Report General Microand portion (8 bytes) of data read; error if data not equal expected data of C.3.

Conditions:

C.5 Report FU Micrand portion of CS data read in C.4; error if data not equal expected data of C.3.

C.6 Read, format and report Control Store parity status; error if not 0.

If C.4, C.5, and C.6 failed or Control Store Location Pointer value equals FF(hex), go to C.7.

If Repeat Iteration set, write value of Control Store Location Pointer to the Micrand Address Register and go to C.3.

Else, increment the Control Store Location Pointer and go to C.2.

C.7 Read and report the contents of the Control Store Micrand Address Register; error if not equal Control Store Location Pointer value + 1.

If C.6 failed, execute C.8 and C.9, else skip C.8 and C.9.

C.8 a. Read 128 bits (16 bytes) of data from Control Store location per CS Micrand Address Register (contains subsequent address from address read by C.4).

b. Report General Micrand portion (8 bytes) of data read; error if not equal General Micrand data written into this location in C.0.

C.9 a. Report FU Micrand portion (8 bytes) of data read in C.8; error if not equal FU Micrand data written into this location in C.0.

If Repeat Iteration set, write the value of the Control Store Location Pointer to the Micrand Address Register and go to C.3.

If Control Store Location Pointer value not equal FF(hex),
increment Control Store Location Pointer and go to C.2.
Else, increment flag.
If flag = CS upper 2K, set DEC bit 46 for upper 2K.
If flag = shadow , set DEC bit 32 for shadow memory.
If flag = done, EXIT SUBSECTION.
ELSE, go to I.1.

Incrementing Pattern (Hexadecimal):

Intended Location	General Micr Portion	FU Micr Portion
000	0000 0000 0000 0000	0000 0000 0000 0000
001	0101 0101 0101 0101	0101 0101 0101 0101
002	0202 0202 0202 0202	0202 0202 0202 0202
|	|	|
|	|	|
0FE	FEFE FEFE FEFE FEFE	FEFE FEFE FEFE FEFE
0FF	FFFF FFFF FFFF FFFF	FFFF FFFF FFFF FFFF
100	0000 0000 0000 0000	0000 0000 0000 0000

Parity Status Format:

See section 24 Description.

Section 24, Subsection 3

This subsection tests the Control Store memory addressing logic both external and internal to the Control Store memory chips.

Method:

The Control Store consists of 1 by 1024-bit memory chips, one chip for each bit position in the memory data. In order to thoroughly test the internal addressing circuitry in each of these 1 by 1024-bit memory chips, a forward March Pattern testing technique with special data patterns is used. This technique also completes the testing of addressing logic external to the memory chips, but still on the Control Store ZIF boards.

The basic test procedure, using a forward March Pattern, is as follows:

a. Write all locations in the Control Store with a base pattern (all locations contain same data).

b. Read one location (Test Address X) and verify that the data has not been altered (equal base pattern) and that no parity error occurred.

c. Write the location just read with a special test data pattern.

Steps b and c are repeated using incrementing test addresses (000 to 7FF(hex)) until a data miscompare and/or parity error is sensed, or all locations have been read and written in a forward direction.

To thoroughly test the 1 by 1024-bit memory chips, steps a through c are then repeated using a different base pattern.

Note that whenever a Control Store location is read, the subsequent location's data is also parity checked. Thus, the location read may contain good data but a parity error is reported because the next location contains bad parity. Because of this, C.7 and C.8 will report and check data from the subsequent Control Store location whenever a parity error is reported by C.6.

Initialization:

I.0 - Clear DEC bit 32.
I.1 - Send Master Clear and Clear Errors to processor.

Repeat the following conditions for each Base Pattern.

Conditions:

C.0 a. Write all locations in Control Store memory with Base Pattern.
b. Clear processor errors.
c. Report Base Pattern number in Condition Result byte 7; informational.

C.1 Report Base Pattern data (8 bytes) which was written to both the General Microrand portion and FU Microrand portion of each Control Store location; informational.

C.2 a. Initialize the Test Address X to zero.
b. Report the Test Pattern data (8 bytes) which will be written to both the General Microrand portion and FU Microrand portion of each Control Store location; informational.

C.3 Report the Test Address X value in Condition Result bytes 6 and 7; informational.

C.4 a. Read Test Address X location from Control Store.
b. Report General Microrand portion of data read from Control Store; error if not equal Base Pattern.

C.5 Report FU Microrand portion of data read from Control Store in C.4; error if not equal Base Pattern.

C.6 Read, format and report the Control Store parity status; error if not 0.

C.7 If C.6 failed read Control Store location X+1 and check and report General Microrand portion of data read; error if data is not equal Base Pattern for addresses 0001 through 7FF(hex), Test Data Pattern for address 000.

If C.6 did not fail, nothing valid to report, so report 0--0.

C.8 If C.6 failed, check and report FU Microrand portion of data read in C.7; error if data is not equal Base Pattern for addresses 001 through 7FF(hex), Test Data Pattern for address 000.

If C.6 did not fail, nothing valid to report, so report 0--0.

If Repeat Iteration set, write Test Address X value to Microrand Address Register and go to C.4.

C.9 Write Test Data Pattern to Test Address X and Clear processor errors. Nothing valid to report so report 0--0.

Increment Test Address X.

If Test Address X value is less than 800(hex) go to C.3.

If current pattern is not equal to Last Base Pattern, go to C.0 and repeat using next Base Pattern.
Else, increment flag.
If flag = CS upper 2K, set DEC bit 46 for upper 2K.
If flag = shadow , set DEC bit 32 for shadow memory.
If flag = done, EXIT SUBSECTION.
Else, go to I.1.

Test Patterns (Hexadecimal):

Test Address X	Test Data Pattern
000	7F7F 7F7F 7F7F 7F7F
001	
Incrementing Address	Same Data Pattern
7FE	
7FF	7F7F 7F7F 7F7F 7F7F

Base Patterns (Hexadecimal):

0 0000 0000 0000 0000
1 8080 8080 8080 8080

Parity Status Format:

See section 24 Description.

SECTION 25 - COMPLETE TEST OF MAC BYTE SELECT LOGIC

This section completes the operational verification of the MAC Byte Count Translation and Byte Selection logic. This logic has been used to some extent in previous sections; however, not all circuit gates were verified as operating correctly.

Subsection 0 - Complete testing of the MAC Byte Select signal formation for Type 4 Soft Control Memories (AC and ALN).

Subsection 1 - Complete testing of the MAC Byte Count Translation logic.

Subsection 2 - Complete testing of the MAC Register File Byte Select Decoder logic.

Hardware Tested:

- The MAC byte 1 select Mux (type 4 leg) and the MAC byte 3 select Mux (type 4 leg), both circuits on the MAC BK ZIF board.
- The MAC Byte Count Decoder logic on the MAC BK ZIF board which feeds the MAC Bus Byte Select logic.
- The MAC Register File Byte Select Decoder logic on the MAC BK ZIF board.

Method:

See individual subsection Method documentation.

Section 25, Subsection 0

This subsection completes testing of the MAC Byte Select signal formation for Type 4 Soft Control Memories (AC and ALN).

Method:

The MAC Byte Select signal formation for Type 4 Soft Control Memories includes special logic which simultaneously writes both byte 0 and byte 1 of the SCM memory location with byte 0 data prior to writing byte 1 with byte 1 data. Similar logic exists to simultaneously write both byte 2 and byte 3 of the SCM memory location with byte 2 data prior to writing byte 3 with byte 3 data. The following test procedure is used to verify this logic's operation:

a. Write location 0 of AC SCM with data containing bad parity in byte 1.

b. Write only one byte of location 0 in AC SCM with good parity data. This good parity data byte should be written into both byte 0 and byte 1 of AC SCM location 0.

c. Sweep location 0 of AC SCM and check parity status to ensure that byte 1 now contains good parity. If byte 1 was not written with byte 0 data, a parity error will be reported.

A similar procedure is used for testing the writing of byte 3 with byte 2 data.

Initialization:

- Send Master Clear to processor.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write AC Soft Control 2 location 0 with following data pattern which will produce parity error in byte 1: pattern equals 00 01 00 00(hex).
c. Clear bit 18 of PTM Register.
d. Write one byte of 00 data to AC Soft Control 2 location 0.
e. Clear errors in processor.
f. Sweep location 0 of AC Soft Control 2.
g. Read and report AC Soft Control 2 parity status - PFS Register 80, bit 45; error if bit 45 is not clear.

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write AC Soft Control 2 location 0 with following data pattern which will produce a parity error in byte 3: pattern equals 00 00 00 01(hex).
c. Clear bit 18 of PTM Register.
d. Write three bytes of 00 data to AC Soft Control 2 location 0.
e. Clear errors in processor.
f. Sweep location 0 of AC Soft Control 2.
g. Read and report AC Soft Control 2 parity status - PFS Register 80, bit 45; error if bit 45 is not clear.

Parity Status Format:

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	AC SC 2 (loc A2E1)	80	45

Section 25, Subsection 1

This subsection completes testing of the MAC Byte Count translation logic.

Method:

The MAC Byte Count translation logic which generates byte select signals for the MAC Byte Selects generates Byte 0 Select when the MAC Byte Count is 0 and 4, Byte 1 Select when the Byte Count is 1 and 5, Byte 2 Select when the Byte Count is 2 and 6 and Byte 3 Select when the Byte Count is 3 and 7. This subsection will verify the proper Byte Select translation for MAC Byte Counts of 4, 5, 6, and 7 using the following method:

a. Start writing AC Soft Control 2 at location 0.

b. Output 12 bytes of error free 00 data.

c. Output 4 bytes of test pattern, one byte with bad parity.

d. Output 4 more bytes of error free 00 data and terminate write.

e. Sweep AC Soft Control 2 locations 0, 1, and 2 and check that no parity error is reported.

f. Sweep AC Soft Control 2 location 3 and verify that a parity error is reported.

This sequence is repeated four times with the test pattern having one bad parity byte in each byte position one time. If the MAC byte Count translation logic does not operate correctly, one of the test patterns will not result in a location 3 parity error.

Initialization

- Send Master Clear to processor.

Repeat the following conditions for each test pattern.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Send write AC Soft Control 2 function to processor.
c. Output 12 bytes of 00 data (good parity).
d. Output 4 bytes of Test Pattern data (one byte contains bad parity).
e. Output 4 bytes of 00 data (good parity) and terminate output.
f. Clear bit 18 of PTM Register.
g. Clear errors in processor.
h. Report Test Pattern number in Condition Result byte 1 and Test Pattern data in Condition Result bytes 4 through 7; informational.

Conditions:

C.1
a. Sweep AC Soft Control 2 locations 0, 1, and 2.
b. Read, format and report PFS Register 80; bit 45; error if bit 45 is not clear.

C.2
a. Sweep AC Soft Control 2 locations 0, 1, 2, and 3.
b. Read, format and report PFS Register 80 bit 45; error if bit 45 is not set.

C.3
a. Write 20 bytes of 00 data to AC SCM PAK 0 beginning at location 0.
b. Clear errors in processor.
c. Sweep AC Soft Control 2 locations 0, 1, 2, 3, and 4.
d. Read, format and report PFS Register 80 bit 45; error if bit 45 is not clear.

Exit subsection if no more test patterns.

Test Patterns (Hexadecimal):

0	01 00 00 00
1	00 01 00 00
2	00 00 01 00
3	00 00 00 01

Parity Status Format:

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	AC SC 2 (loc A2E1)	80	45

Section 25, Subsection 2

This subsection completes testing of the MAC Register File Byte Select logic.

Method:

The MAC Register File Byte Select Decoder generates dual byte selects for the following conditions: Select bytes 0 and 1 when byte count equal 0, select bytes 2 and 3 when byte count equal 2, select bytes 4 and 5 when byte count equal 4, and select bytes 6 and 7 when byte count equal 6. This subsection will verify the proper byte selects for byte counts of 0, 2, 4, and 6 using the following basic procedure:

Method:

a. Write one byte of data (FF(hex)) to byte 0 of Register File location 00. This byte should be entered into both byte 0 and byte 1 of location 00.

b. Read Register File location 00. Data read should contain FF(hex) in both byte positions 0 and 1.

This procedure is repeated by writing into byte 2, then byte 4, and finally byte 6.

Initialization:

- Master Clear processor.
- Clear errors in processor.
- Write Register File location 0 with 8 bytes of 0 data.

Conditions:

C.0 a. Write Register File location 0 with 1 byte of data; write data equals FF(hex).
b. Report write data equal to FF 00 00 00 00 00 00 00(hex); informational.

C.1 Read and report 8 bytes of data from Register File location 0; error if not equal to FFFF 0000 0000 0000(hex).

C.2 a. Write Register File location 0 with 3 bytes of data; write data equals 00 00 FF(hex).
b. Report write data equal to 00 00 FF 00 00 00 00 00(hex); informational.

C.3 Read and report 8 bytes of data from Register File location 0; error if not equal to 0000 FFFF 0000 0000(hex).

C.4 a. Write Register File location 0 with 5 bytes of data; write data equals 00 00 00 00 FF(hex).
b. Report write data equal to 00 00 00 00 FF 00 00 00(hex); informational.

C.5 Read and report 8 bytes of data from Register File location 0; error if not equal to 0000 0000 FFFF 0000(hex).

C.6 a. Write Register File location 0 with 7 bytes of data; write data equals 00 00 00 00 00 00 FF(hex).
b. Report write data equal to 00 00 00 00 00 00 FF 00(hex); informational.

C.7 Read and report 8 bytes of data from Register File location 0; error if not equal to 0000 0000 0000 FFFF(hex).

SECTION 26 - COMPLETE TEST OF MAC TYPE DECODE LOGIC

This section completes the operational verification of the MAC Type Decode logic. This logic has been used to some extent in previous sections; however, not all circuit gates were verified as operating correctly.

Subsection 0 - Complete testing of the MAC Type Decode logic.

Hardware Tested:

- The Type Decode logic on the MAC BZ ZIF board, and the corresponding type signal fanouts.
- The Type Decode logic on the MAC BK ZIF board, and the corresponding type signal fanouts.

Method:

See the individual subsection Method documentation.

Section 26, Subsection 0

This subsection completes testing of the MAC Type Decode logic.

Method:

This subsection verifies that the MAC Type Decode logic circuits do not generate more than one set of type signals for any one type code (e.g. type code is equal to 4, decoder sending out both Type 4 and Type 5 signals is a failure).

To verify that no more than one type signal is generated for any one type code, the subsection writes one type memory/registers with a bad parity data pattern. Then one location or register from each type memory/register group is read or swept. Only the memory/register type written with bad parity should report a parity error. All other memory/register types should be parity error free. If the type decode logic is generating more than one type signal at a time, additional memory/register types will report parity errors.

Initialization:

- Send Master Clear and Clear Errors to processor.
- Write all memory/register type units with 8 bytes of 0's:
  Type 0, Control Word 1 equals zero, Control Word 2 equals 31(hex) (CS Microand Address Reg).
  Type 1, Control Word 1 equals zero, Control Word 2 equals 31(hex) (Control Store location 0).
  Type 3, Control Word 1 equals zero, Control Word 2 equals 31(hex) (Reference ROM locations 31(hex) and 32(hex)).
  Type 4, Control Word 1 equals zero, Control Word 2 equals 31(hex) (AC SC 0, locations 31(hex) and 32(hex)).
  Type 5, Control Word 1 equals zero, Control Word 2 equals 31(hex) (BDP Port A Board, locations 31(hex) and 32(hex)).
  Type 6, Control Word 1 equals zero, Control Word 2 equals 31(hex) (IF C180 Decode location 31(hex) and 32(hex)).
  Type 7, Control Word 1 equals zero, Control Word 2 equals 31(hex) (Register File location 31(hex)).

NOTE

AC SC 0 does not physically exist but is still written to in order to test Decoder.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write bad parity 8 byte Test Pattern to Type 0 with Control Word 1 equal to zero, Control Word 2 equal to 31(hex); Control Store Microand Address Register.
c. Clear bit 18 of PTM Register and Clear errors in processor.
d. Read/sweep all memory/register types using Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Subroutine).
e. Read, format and report Parity Status; error if PFS Register 86, bit 3 is not set and all other bits are not clear.

C.1
a. Write 8 bytes of good parity 0 data to Type 0, Control Word 1 equals zero, Control Word 2 equals 31(hex).
b. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
c. Write bad parity 8 byte Test Pattern to type 1 with Control Word 1 equal to zero, Control Word 2 equal to 31(hex); Control Store location per Address Register.
d. Clear bit 18 of PTM Register and clear errors in processor.
e. Read/sweep all memory/register types using Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Subroutine).
f. Read, format and report Parity Status; error if PFS Register 86, bits 17 and 23 are not set and all other bits are not clear.

Conditions:

C.2
a. Write 8 bytes of good parity 0 data to Type 1, Control Word 1 equal to zero, Control Word 2 equal to 31(hex).
b. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
c. Write bad parity 8 byte Test Pattern to type 3 with Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Reference ROM locations 31(hex) and 32(hex)).
d. Clear bit 18 of PTM Register and clear errors in processor.
e. Read/sweep all memory/register types using Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Subroutine).
f. Read, format and report Parity Status; error if PFS Register 86, bit 38 is not set and all other bits are not clear.

C.3
a. Write 8 bytes of good parity 0 data to Type 3, Control Word 1 equal to zero, Control Word 2 equal to 31(hex).
b. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
c. Write bad parity 8 byte Test Pattern to type 4 with Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (AC SC 0 - nonexistant).
d. Clear bit 18 of PTM Register and clear errors in processor.
e. Read/sweep all memory/register types using Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Subroutine).
f. Read, format and report Parity Status; error if any PFS bits set.

C.4
a. Write 8 bytes of good parity 0 data to Type 4, Control Word 1 equal to zero, Control Word 2 equal to 31(hex).
b. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
c. Write bad parity 8 byte Test Pattern to type 5 with Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (BDP Port A board, locations 31(hex) and 32(hex)).
d. Clear bit 18 of PTM Register and clear errors in processor.
e. Read/sweep all memory/register types using Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Subroutine).
f. Read, format and report Parity Status; error if PFS Register 81, bit 4 is not set and all other bits are not clear.

C.5
a. Write 8 bytes of good parity 0 data to Type 5, Control Word 1 equal to zero, Control Word 2 equal to 31(hex).
b. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
c. Write bad parity 8 byte Test Pattern to type 6 with Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (IF C180 Decode locations 31(hex) and 32(hex)).
d. Clear bit 18 of PTM Register and clear errors in processor.
e. Read/sweep all memory/register types using Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Subroutine).
f. Read, format and report Parity Status; error if PFS Register 86, bit 50 is not set and all other bits are not clear.

Conditions:

C.6
a. Write 8 bytes of good parity 0 data to Type 6, Control Word 1 equal to zero, Control Word 2 equal to 31(hex).
b. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
c. Write bad parity 8 byte Test Pattern to type 7 with Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Register File location 31(hex)).
d. Clear bit 18 of PTM Register.
e. Read/sweep all memory/register types using Control Word 1 equal to zero, Control Word 2 equal to 31(hex) (Subroutine).
f. Read, format and report Parity Status; error if PFS Register 84, bits 57 and 58 are not set and all other bits are not clear.

Write 8 bytes of good parity 0 data to Type 7, Control Word 1 equal to zero, Control Word 2 equal to 31(hex); and clear errors in processor.

Test Pattern (Hexadecimal):

0 00 01 00 00 00 00 00 01

Parity Status Format:

Condition Result Bit	Parity Error At		PFS Register No.	Bit
63	CS Micr Address Register -Type 0	(loc C1C)	86	3
62	CS FU Micr Register byte 7 - Type 1	(LSI C2G14)	86	23
61	CS FU Micr Register byte 0 - Type 1	(LSI C2G02)	86	17
60	MAC Reference ROM data - Type 3	(loc C1D)	86	38
59	ALN SC	(loc C2K1)	80	44
58	BDP Port A Board - Type 5	(loc A3B1)	81	4
57	IF C180 Decode PAK1 - Type 6	(loc C2A1)	86	50
56	Register File bits 32--63-Type 7	(loc B3A2)	84	58
55	Register File bits 0--31-Type 7	(loc B3A1)	84	57

Read/Sweep Subroutine

- Read 8 bytes of data from Type 0, Control Word 1 equals zero, Control Word 2 equals 31(hex).
- Read 8 bytes of data from Type 1, location 0.
- Read 2 bytes of data (sweep) from Type 3, Control Word 1 equals zero, Control Word 2 equals 31(hex).
- Read 2 bytes of data (sweep) from Type 4, Control Word 1 equals zero, Control Word 2 equals 31(hex).
- Read 2 bytes of data (sweep) from Type 5, Control Word 1 equals zero, Control Word 2 equals 31(hex).
- Read 2 bytes of data (sweep) from Type 6, Control Word 1 equals zero, Control Word 2 equals 31(hex).
- Read 8 bytes of data from Type 7, Control Word 1 equals zero, Control Word 2 equals 31(hex).

SECTION 27 - COMPLETE TEST OF MAC GO DECODER

This section completes the operational verification of the MAC Go Decoders fed by the Partial Address Incrementer. This logic has been used to some extent in previous sections; however, not all circuit gates were verified as operating correctly.

Subsection 0 - This subsection completes testing of the MAC Soft Control Go Decoder circuits involved with generation of the Go Soft Control Memory signals.

Subsection 1 - This subsection completes testing of the MAC BDP Control Memory Go Decoder circuits involved with generation of the Go BDP memory signals.

Hardware Tested:

- The MAC Soft Control Go Decoder logic which generates Go AC SC 1, Go AC SC 2 and Go ALN SC signals (MAC BZ ZIF board).
- The MAC BDP Control Memory Go Decoder logic which generates Go BDP Memory 0, Go BDP Memory 1, Go BDP Memory 2, Go BDP Memory 3 and Go BDP Memory 4 signals (MAC BZ ZIF board)
- The logic involved with advancing the MAC Partial Address Incrementer (both MAC ZIF boards - BZ and BK).

Method:

Both subsections use the same basic testing method.

The AC/ALN SC memories (Type 4) are addressed using Control Word 2 to provide both the proper memory selection (AC Soft Control 2, AC Soft Control 1, ALN SC) and the address within the memory selected. To test the Go Decoder, one location within one of the Type 4 memories is written with bad parity data. Then all locations in all of the Type 4 memories are swept. Only one of the Type 4 memories should report a data parity error, the one that had bad parity data written into it. If the Go Decoder is faulty, more than one of the Type 4 memories will report a data parity error. This process is repeated with different address combinations in order to thoroughly test the Go Decoder generation of AC/ALN SC Go signals.

The BDP memories (Type 5) are addressed using Control Word 1 and Control Word 2 to provide both the proper memory selection and the address within the memory selected. This is similar to AC/ALN SC memories as described above. Therefore the same testing technique is used to test the Go Decoder for BDP memory Go signal generation.

To include testing the Partial Address Incrementer advancing logic, a number of good parity data words are output to the memories followed by the output of a bad parity data word. The transmittion of the good parity data will cause the Partial Address Incrementer to advance across memory selection boundaries. The bad parity data word will then be written into a particular memory of the same type. When the memories are swept, only the particular memory should report a parity error. If the advancing logic is malfunctioning the wrong memory will report a parity error.

Parity Status Format:

Same for all subsections.

Condition Result Bit	Parity Error At	PFS Register No.	Bit
63	ALN SC (loc C2K1)	80	44
62	AC SC 1 (loc A2G1)	80	46
61	AC SC 2 (loc A2E1)	80	45
60	Not connected to anything		
59	BDP RAM PAK 0 (loc A3A2)	81	11
58	BDP RAM PAK 0 (loc A3A2)	81	9
57	BDP Port B PAK (loc A3C1)	81	5
56	BDP Port A PAK (loc A3B1)	81	4
55	BDP RAM PAK 1 (loc A3B2)	81	3

Section 27, Subsection 0

This subsection completes testing of the MAC Soft Control Go Decoder circuits involved with generation of the Go Soft Control Memory signals

Method:

See section 27 Description Method.

Initialization:

- Send Master Clear to processor

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 8-byte bad parity Test Pattern to Type 4 memories, Control Word 1 equals zero, Control Word 2 equals zero.
c. Clear bit 18 of PTM Register.
d. Clear errors in processor.
e. Sweep all Type 4 memories' locations.
f. Read, format and report Parity Status; error if not 0. AC SC 0 does not exist, so cannot cause parity error.

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 40(hex) 4-byte words of 00 data to Type 4 memories with Control Word 1 equals zero, Control Word 2 equals zero.
c. Output 8-byte bad parity Test Pattern (continuation of step b write function data transmission).
d. Clear bit 18 of PTM Register.
e. Clear errors in processor.
f. Sweep all Type 4 memories' locations.
g. Read, format and report Parity Status; error if PFS Register 80 bit 45 is not set and all other bits are not clear.

C.2
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 80(hex) 4-byte words of 00 data to Type 4 memories with Control Word 1 equal to zero, Control Word 2 equal to zero.
c. Output 8-byte bad parity Test Pattern (continuation of step b write function data transmission).
d. Clear bit 18 of PTM Register.
e. Clear errors in processor.
f. Sweep all Type 4 memories' locations.
g. Read, format and report Parity Status; error if PFS Register 80 bit 46 is not set and all other bits are not clear.

C.3
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write C0(hex) four byte words of 00 data to Type 4 memories with Control Word 1 equal to zero, Control Word 2 equal to zero.
c. Output 8-byte bad parity Test Pattern (continuation of step b write function data transmission).
d. Clear bit 18 of PTM Register.
e. Clear errors in processor.
f. Sweep all Type 4 memories' locations.
g. Read, format and report Parity Status; error if PFS Register 80 bit 44 is not set and all other bits are not clear.

Write all Type 4 memories' locations with good parity 00 data and clear errors in processor.

Test Pattern (Hexadecimal):TDSTAMP.
0 00 00 00 01 00 00 00 00

Parity Status Format:

See section 27 Description.

Section 27, Subsection 1

This subsection completes testing of the MAC BDP Control Memory Go Decoder circuits involved with generation of the Go BDP memory signals.

Method:

See section 27 Description Method.

Initialize by sending Master Clear to processor.

Conditions:

C.0
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 8 byte bad parity Test Pattern to Type 5 memories, Control Word 1 equals zero, Control Word 2 equals zero.
c. Clear bit 18 of PTM Register.
d. Clear errors in processor.
e. Sweep all Type 5 memories' locations.
f. Read, format and report Parity Status; error if PFS Register 81 bit 4 is not set and all other bits are not clear.

C.1
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 100(hex) 4-byte words of 00 data to Type 5 memories with Control Word 1 equals zero, Control Word 2 equals zero.
c. Output 8-byte bad parity Test Pattern (continuation of step b write function data transmission).
d. Clear bit 18 of PTM Register.
e. Clear errors in processor.
f. Sweep all Type 5 memories' locations.
g. Read, format and report Parity Status; error if PFS Register 81 bit 5 is not set and all other bits are not clear.

C.2
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 200(hex) 4-byte words of 00 data to Type 5 memories with Control Word 1 equal to zero, Control Word 2 equal to zero.
c. Output 8-byte bad parity Test Pattern (continuation of step b write function data transmission).
d. Clear bit 18 of PTM Register.
e. Clear errors in processor.
f. Sweep all Type 5 memories' locations.
g. Read, format and report Parity Status; error if PFS Register 81 bit 9 is not set and all other bits are not clear.

Conditions:

C.3
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 300(hex) 4-byte words of 00 data to Type 5 memories with Control Word 1 equal to zero, Control Word 2 equal to zero.
c. Output 8-byte bad parity Test Pattern (continuation of step b write function data transmission).
d. Clear bit 18 of PTM Register.
e. Clear errors in processor.
f. Sweep all Type 5 memories' locations.
g. Read, format and report Parity Status; error if PFS Register 81 bit 11 is not set and all other bits are not clear.

C.4
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 400(hex) 4-byte words of 00 data to Type 5 memories with Control Word 1 equal to zero, Control Word 2 equal to zero.
c. Output 8-byte bad parity Test Pattern (continuation of step b write function data transmission).
d. Clear bit 18 of PTM Register.
e. Clear errors in processor.
f. Sweep Type 5 memories' locations from 0 through 3FF(hex).
g. Read, format and report Parity Status; error if all bits are not clear.

C.5
a. Sweep one Type 5 memory location at Control Word 1 equals 4, Control Word 2 equals zero.
b. Read, format and report Parity Status; error if PFS Register 81 bit 3 is not set and all other bits are not clear.

C.6
a. Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
b. Write 500(hex) 4-byte words of 00 data to Type 5 memories with Control Word 1 equal to zero, Control Word 2 equal to zero.
c. Output 8-byte bad parity Test Pattern (continuation of step b write function data transmission).
d. Clear bit 18 of PTM Register.
e. Clear errors in processor.
f. Sweep Type 5 memories' locations from 0 through 4FF(hex).
g. Read, format and report Parity Status; error if all bits are not clear.

C.7
a. Sweep one Type 5 memory location at Control Word 1 equals 5, Control Word 2 equals zero.
b. Read, format and report Parity Status; error if PFS Register 81 bit 3 is not set and all other bits are not clear.

Write all Type 5 memories' locations with good parity 00 data and clear errors in processor.

Test Pattern (Hexadecimal):

0 00 00 00 01 00 00 00 00

Parity Status Format:

See section 27 Description.

SECTION 28 - MISCELLANEOUS MAC LOGIC TESTING

This section tests miscellaneous logic in the MAC that has not been previously tested.

Subsection 0 - Test MAC Data Mux 4 and complete the testing of MAC Data Mux 5.

Subsection 1 - Perform a partial test of MAC Data Mux 1 and Data Mux 2.

Subsection 2 - This subsection tests that a Master Clear Function to processor MAC, generates Soft Control Go signals to the AC and ALN Soft Control Memories, and function signal clears MAC Address Incrementer.

Hardware Tested:

- All four legs of Data Mux 4 on the MAC BK ZIF board.
- The Data Mux 4 Parity Generator, for the limited data available through Data Mux 4 (data is hardwired) - BK ZIF board.
- The Mux 4 path and Element ID path on Data Mux 5 (MAC BK ZIF board).
- This subsection completes testing of the MAC Address Decoder logic on the MAC BK ZIF board.
- Partial test of Data Mux 1, Data Mux 1 Parity Generator and Data Mux 2; all on the MAC BZ ZIF board. The portion of this logic not tested here will be tested by later executed processor Identification Tests.
- Master Clear forcing AC and ALN Soft Control Go signals on MAC BZ ZIF board.

Method:

See the individual subsection Method documentation.

Section 28, Subsection 0

This subsection tests MAC Data Mux 4 and completes the testing of MAC Data Mux 5.

Method:

This subsection tests Mux 4 by reading Processor Registers 10(hex) (Element ID), 11(hex) (Maintenance ID) and 12(hex) (Options Installed Register) and comparing the data read to the preset contents of Test Parameters 9 and 10. The Parity Status from PFS Register 86 (bits 32 and 35) is also read and verified that proper parity was generated for the read data.

Initialization:

- None.

Conditions:

C.0 a. Master clear and clear error P3.
b. Read Element ID (EID) and:

If PARAM9 is FFFF(hex), report serial number read from Element ID register. If PARAM19 is not FFFF(hex), report error if content of PARAM9 and serial number read from EID are not equal.

C.1 Report model number read from EID register:

Machine	Model Number
855	30
845	31
860	32
850	33

C.2 Read, format and report Parity Status; error if not zero.

C.3 a. Clear P3.
b. Read Maintenance ID register (MID) and:

If PARAM10 bits 2**15 through 2**8 is FF(hex), report value read from MID. If testing CPU-B, report an error if bit 63 is not set.

If PARAM10 bits 2**15 through 2**8 is not FF(hex), report error if this number and the value read from MID are not equal.

C.4 Read, format and report Parity Status; error if not zero.

C.5 a. Clear P3 error.
b. Read P3 Option Installed register (OI) and:

If PARAM10 bits 2**7 through 2**0 is FF(hex), report value read from OI. If testing CPU-B, report an error if bit 59 is not set.

If PARAM10 bits 2**7 through 2**0 is not FF(hex), report error if this number is not equal to the value read from OI.

C.6 Read, format and report Parity Status; error if not zero.

If testing CPU-1, continue; Else, EXIT SUBSECTION

C.7 Read and report contents of CPU-0 Options Installed Register. Error if bit 59 (dual cpu) is not set.

Exit Subsection

Section 28, Subsection 1

This subsection performs a partial test of MAC Data Mux 1 and Data Mux 2.

Method:

This subsection performs a partial test of MAC Data Mux 1 and Mux 2 by reading Processor Register 00 (Status Summary Register) and comparing the data read to known conditions in the processor. The parity status from PFS Register 86 (bit 32) is also read and verified that proper parity was generated for the read data.

Since the Status Summary Register contains bits that cannot be controlled this early in the processor testing sequence, only some of the bits are checked. Later executed processor Identification Tests will complete the testing of the Status Summary Register bits and thus complete testing of Data Mux 1 and Mux 2.

Initialization:

- Send Master Clear and Clear Errors to processor.

Conditions:

C.0 Read and report 8 bytes from Processor Register number 00, Status Summary Register; error if both Environmental Warning bits are not clear and Processor Halted bit is not set. Other bit values are unknown.

C.1 Read and report PFS Register 86 bit 32; error if bit is not clear.

Processor Register 00 (Status Summary) Format (each byte the same):

```
bit n
  |      Not used - Should always be 0
bit n+1
bit n+2 - C180 Monitor Mode
bit n+3 - Short Environmental Warning (power warning)
bit n+4 - Processor Halted
bit n+5 - Uncorrected Error
bit n+6 - Corrected Error
bit n+7 - Long Environmental Warning (physical environment warning)
```

Section 28, Subsection 2

This section tests that a Master Clear Function to MAC generates Soft Control Go signals to the AC and ALN Soft Control Memories, and function signal clears MAC Address Incrementer.

Method:

Bad parity data is written into location 0 of each AC and ALN Soft Control RAM, followed by writing good parity data into location 1 of each SC RAM. This leaves the RAM address registers in each Soft Control set to 01 which causes good parity data to be gated from the memories to their read data parity checkers.

The MAC Address Incrementer Register is set to FF(hex) by initiating a Sweep function of BDP Port A RAM without inputting the first byte.

A Master Clear Function is then sent to MAC. This function should clear the Address Incrementer Register and cause a GO SC signal to all of the AC and ALN Soft Control Memories. The AC and ALN Soft Control Memories will respond to the GO SC signals by each reading location 0 which will result in data parity errors being reported for each AC and ALN Soft Control (assumes that the hardware is not faulty).

Initialization:

- Send Master Clear to processor.
- Set bit 18 of PTM Register (A0) to force 0 parity bit on data through MAC.
- Write 8 bytes of data to each AC and ALN Soft Control. Data equals 00 00 00 01 00 00 00 00(hex) (location 0 of each SC will contain bad parity data and location 1 will contain good parity data).
- Clear bit 18 of PTM Register.

Conditions:

C.0
a. Clear errors in processor.
b. Read, format and report Parity Status from all AC and ALN Soft Control Memories; error if not 0.

C.1
a. Send Read Function to MAC for BDP Port A RAM location FF(hex) (Op Code equals 4, Type Code equals 5, Control Word 1 equals zero, Control Word 2 equals FF(hex)). Do not input any data.
b. Send Master Clear Function to MAC.
c. Read, format and report Parity Status from all AC and ALN Soft Control Memories; error if all bits are not set.

Write good parity data to each AC and ALN Soft Control Memory location 0.

Conditions:

Parity Status Format:

Condition Result Bit	Parity Error At	PFS Register No.	PFS Register Bit
63	AC SC PAK 2 (loc A2G1)	80	46
62	AC SC 2 (loc A2E1)	80	45
61	ALN SC PAK (loc C2K1)	80	44

SECTION 29 - TEST FFC LOGIC

The purpose of this section is to test the First Failure Capture logic of PIP3. This logic is designed to record, via a status bit, the first processor fault status register that collects an error. This logic feature is enabled by DEC bit 43.

Subsection 00 - This subsection verifies that an error logged will set the first failure status bit in register 87 properly.

Hardware tested -

Method - Refer to the Method documentation provided with the respective subsection for detailed information.

Section 29, Subsection 0

This subsection tests the first failure capture feature of the 960 error logs.

Conditions:

C.0 a. Write all delay registers with zero data. Clear CPU errors. Read and report clock parity status. Error if nonzero.

C.1 a. Report the currently selected clock. (information only)

C.2 a. Write data with a parity error in one of the clock delay registers on the selected clock array. Report channel status. (Error if nonzero)

C.3 a. Clear CPU errors. Read and report clock parity status. (Error if not as expected.)

C.4 a. Attempt to select a clock using an address with a parity error. This will force a parity error report from all the clock arrays. Read and report clock parity status. (Error if all clock arrays have not reported an error.)

C.5 a. Read and report the first failure bits. (Error if not as expected).

C.6 a. Write zeroes to the selected clock array to clear the parity errors. Report channel status. (error if nonzero)

Repeat from C.1 for all clock ID's.
End subsection.

Conditions:

iteration	clock
0	F
1	E
2	3
3	5
4	4
5	6
6	7
7	8
8	9
9	B
A	C
B	0
C	1
D	2
E	A
F	D

Section 29, Subsection 01

This subsection restores the clock array data read initially in the previous subsection.

The data read in the previous subsection will be written to the appropriate arrays. It will be read out of the clocks again and verified with the original data. Finally it will be written back to the clock arrays.

Conditions :

C.0 a. Write the saved clock data back to the appropriate clock arrays.
b. Read the clock delay data.
c. Write the clock data to the appropriate clock array.
d. Verify the data read in step b.
e. If no errors and not all clock arrays initialized, repeat from a. for the next clock array.
f. Report data read in b. (Error if data miscompare detected.)

C.1 a. Report ID of last clock referenced. (Informational)

End subsection.

Abs

Absolute.

AC or A/C

Address Control.

ACMIC

Address Control Functional Microand.

ADATB/RDSB/B

Refers to B Operand data in Multiply/ Divide unit test MDT3/P.

ADATC/RDSC/C

Refers to C Operand data in Multiply/ Divide unit test MDT3/P.

Adder

Refers to Large Adder in ALN test ANT3/P. Refer to Multiply Final Adder in test MDT3/P.

Adder Bit 47 FF

Large Adder Result bit 47 flip-flop in ALN test ANT3/P.

Adrs

Address.

ALN

Arithmetic and Logical Network.

ALU

Arithmetic and Logical Unit.

AOR

Address-Out-of-Range.

Arith

Arithmetic.

ASCII

American Standard Code for Informa- tion Interchange.

ASE

Address Specification Error.

ASID

Active Segment Identifier.

Assy

Assembly.

Aux

Auxiliary.

B

Refers to the operand input to ALN on the RSDB data path. Used in ALN test ANT3/P.

BCD

Binary Coded Decimal.

BD

Branch Delta.

BDP

Business Data Processor.

Bfr

Buffer.

Bin

Binary.

Bkpt

Breakpoint.

1BN

Byte Number.

Bound

Boundary.

Br

Branch.

C

Refers to the operand input to the ALN on the RDSC data path. Used in ALN test ANT3/P.

CA

Condition Action.

CABR

Cache Address Buffer Register.

CAE

Condition Action Extension.

CAR

Cache Address Register.

CBP

Code Base Pointer.

CCR

Clock Condition Register.

CEL

Corrected Error Log.

C.GT.B

C Greater than B signal from the SKG in ALN test ANT3/P.

Chan

Channel.

Clk

Clock.

Clr

Clear.

CM

Central Memory.

CMC

Central Memory Control.

Cnt

Count.

Cntr

Counter.

Coef

Coefficient.

Com

Common.

Comp

Complement.

Cond

Condition.

Confl

Conflict.

Cont

Control.

Conv

Convert, Converter.

Cor

Corrected.

CPU

Central Processing Unit.

ICS

Condition Sensing.

CSA

Control Store Address.

CST

Control Store.

CW1

Control Word 1.

CW2

Control Word 2.

DAI

Data Interchange.

DCDR

Decoder.

DCI

Data control information. Two bytes of information that give the maintenance register address.

DEC

Dependent Environment Control, Decimal.

Decr

Decrement.

Descr

Descriptor.

Destn

Destination.

Disassy

Disassembly.

Div

Divide.

Divisor Rgtr

Divide network register internal to LSI CI divide arrays.

Dlyd

Delayed.

DMR

Debug Mask Register.

Dsbl

Disable.

DSS

Data Subfunction Select Microrand Field.

EBCDIC

Extended Binary Coded Decimal Inter- change Code.

ECC

Error Correction Code.

ECR

C170 Exit Mode Condition Register.

EI

Environment Interface.

EMMR or EMR

Exit Mode Mask Register.

Enbl

Enable.

Encdr

Encoder.

ESE

Environment Specification Error.

EXCH

Exchange.

Exp

Exponent.

Exponent Aux

Exponent Auxiliary board in ALN.

Ext

External.

Fctn

Function.

FF

Flip-flop.

FIFO

First In/First Out.

FLC

C170 Mode Central Memory Field Length.

FNO

Fanout.

FRC

Free Running Counter.

FU

Functional Unit.

Gen

Generator.

Genl

General.

GME

Used by test CTT3/P to indicate a field (bits 62 and 63) within the general micrand.

Hldg

Holding.

i

Operand Register Designator.

IACY

C170 Parcel Select Multiplexer (IFT3/P test).

IAHY

Instruction Assembly Hybrid Multi- plexer (IFT3/P test).

IAPL

C180 Parcel Select Multiplexer (IFT3/P test).

IB02

Instruction Buffer Rank 02 (IFT3/P test).

IB12 valid

Used in ALN test ANT3/P. Instruction Buffer Rank 12 valid signal from Instruction Fetch. This goes active when IF has finished fetching the next instruction from CM.

ICC

Instruction Completion Control.

ICP

Instruction Control Pipeline.

IDS

Immediate Data Select Micrand Field.

IDX

Indexed.

IF

Instruction Fetch.

II

Instruction Issue.

Immed

Immediate.

Incr

Increment.

Indef

Indefinite.

Inf

Infinite.

Inh

Inhibit.

Inp

Input.

Instr

Instruction.

Int

Internal.

Integer Mux

Integer/Exponent Multiplexer in ALN.

Intrpt

Interrupt.

IOU

Input/Output Unit.

ISE

Instruction Specification Error.

IU

Instruction Unit.

J

Operand Register Designator.

JPS

Job Process State.

k

Operand Register Designator.

K

170 18-Bit Immediate Operand.

K/L

Key/Lock.

Kypt

Keypoint.

l

BDP Operand Length.

Ld

Load.

LM

Local Memory.

LMMIC

Local Memory Functional Microand field.

LOS

Loss of Significance.

LRR

Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P, and ICT3/P to indicate a live register read.

LRU

Least Recently Used.

LRW

Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P and ICT3/P to indicate a live register write.

SD

Least Significant Digit.

LSI

Large Scale Integration.

LSM

Last Symbol Mask.

LSMPRE

LOAD/STORE Multiple Address Adder.

Lwr

Lower.

MAC

Maintenance Access Control.

Maint

Maintenance.

MAR

Microand Address Register.

MBE

Multiple Bit Error.

MC

Master Clear.

MCH

Maintenance Channel.

MCMR

Monitor Condition Mask Register. Also refer to MMR.

MCR

Monitor Condition Register.

Mem

Memory.

Micr

Microand.

Microand Constant

Contents of ALN FU Microand bits 0 through 6.

MIS

Miscellaneous Microand Field.

Misc

Miscellaneous.

MMR

Monitor Mask Register.

MOP

Micro operation.

MPS

Monitor Process State.

MR

Maintenance Register.

MSB

Most Significant Bit.

MSC

Microand Sequence Control. A field (bits 0 through 15) within the general microand.

MSD

Most Significant Digit.

MSNZB

Most Significant Nonzero Byte.

Mult

Multiply.

Mux

Multiplexer.

Neg

Negative.

Norm

Normalize.

NPGK

New P Global Key.

NPLK

New P Local Key.

ns

Nanosecond.

o

BDP Operand Address.

Op

Operand.

Opcode

Operation Code.

OPGK

Old P Global Key.

OPLK

Old P Local Key.

OPI

Operand Issue.

Opn

Operation.

OSB

Operating System Bounds.

Out

Output.

Ovfl

Overflow.

P

Program Address.

Par

Parity.

PDM

Processor Detected Malfunction.

PE

Parity Error.

PFA

Page Frame Address.

PFS

Processor Fault Status.

Ph

Phase.

PIT

Process Interval Timer.

PMF

Performance Monitoring Facility.

PN

Page Number.

PO

Page Offset.

PONR

Point Of No Return.

Pos

Positive.

PP

Peripheral Processor.

Prev

Previous.

Pri

Priority.

Prod

Product.

PSM

Page Size Mask.

PSWF

Page Table Search Without Find.

PTA

Page Table Address.

PTD

Page Table Descriptor.

PTL

Page Table Length.

PTM

Processor Test Mode.

PVA

Process Virtual Address.

PW

Partial Write.

Q

C180 16-Bit Immediate Operand.

Quad

Quadrant.

Quotient Rgtr

Divide network register internal to the LSI CI arrays.

R/W

Read/Write.

RAC

C170 Mode Central Memory Reference Address.

RAM

Random Access Memory.

RCL

Read and Clear Lock.

Rcvr

Receiver.

RDS

Register/Data Select.

RDSA

In ALN test ANT3/P, refers to the micrand field that provides the register file address on a write of ALN data and to the data path through DAI for that data.

RDSB

In ALN test ANT3/P, refers to the following data path to ALN. The specified register file is output to the RDSB bus, transferred to the ALN on ADATB, and is passed through the multiple/divide unit and output multiplexer to the general network.

RDSC

In ALN test ANT3/P, refers to following data path. The specified register file is sent to the RDSC bus, trans- ferred to ADATC, then goes to the general network.

Ref

Reference.

Regen

Regenerator.

Rel

Relative.

Rem

Remainder.

Req

Request.

Resync

Resynchronize, Resynchronizer, Resynchronization.

RF

Register File.

RFOx

Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P to identify the register file being used. x is 5, 6, 7, 8, 9, A, B, C, D, E or F (Hexadecimal).

Rgtr

Register.

RMA

Real Memory Address.

ROM

Read Only Memory.

RSL

Read and Set Lock.

RS64

Right Shift 64 signal from the SKG.

Rxx

Identifies ranks in IF and ICP in Test CTT3/P.

SBE

Single Bit Error.

SC

Soft Control.

SCM

Soft Control Memory.

SCT

Special Character Table.

SECDED

Single Error Correction/Double Error Detection.

Seg

Segment.

Sel

Select.

Sep

Separate.

Seq

Sequence.

Shift Mux

Shift Network Multiplexer.

Shifter

Shift Network.

Sig

Significant.

SIT

System Interval Timer.

SKG

Shift Count Generator.

SM

Segment Map.

SMMIC

Segment Map Functional Microrand Field.

SMO(x)

Segment Descriptor Multiplexer Output bit x.

SMPDM

Segment Map Processor Detected Malfunction.

SN

Segment Number.

Spec

Specification.

SPID

Segment/Page Identifier.

STA

Segment Table Address.

STL

Segment Table Length.

Str

Stream.

Subtr

Subtract.

Suppr

Suppression.

SV

Specification Value.

SVA

System Virtual Address.

Sw

Switch.

Syn

Syndrome.

Sync

Synchronize.

Termn

Terminate.

TFA

Tag File Address.

ubit

Microrand Bit.

UCEL1

Uncorrected Error Log 1.

UCEL2

Uncorrected Error Log 2.

UCR

User Condition Register.

UMR

User Mask Register.

Unbr

Unbranch.

Uncond

Unconditional.

Uncor

Uncorrectable.

Undfl

Underflow.

Unlog

Unlogged.

Upr

Upper.

usec

Microsecond.

UTP

Untranslatable Pointer.

Ux

Used in Multiply/Divide test MDT3/P to identify Major Cycle Control bits 0 through 14.

VMID

Virtual Machine Identifier.

Vx

Used in Multiply/Divide test MDT3/P to identify Minor Cycle Control bits 0 through 15.

DS

Write Data Select.

WOI

Word of Interest.

Xfer

Transfer.

Xltr

Translator.

Xmtr

Transmitter.

XOR

Exclusive OR.

ZIF

Zero Insertion Force.

ZF

Zero Flag.

CONTROL DATA

4201 North Lexington Avenue Saint Paul, Minnesota 55126-6198